

Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention

Journal of Educational Computing
Research
0(0) 1–23

© The Author(s) 2018

Reprints and permissions:

sagepub.com/journalsPermissions.nav

DOI: 10.1177/0735633118757015

journals.sagepub.com/home/jec



Wanli Xing¹ and Dongping Du²

Abstract

Massive open online courses (MOOCs) show great potential to transform traditional education through the Internet. However, the high attrition rates in MOOCs have often been cited as a scale-efficacy tradeoff. Traditional educational approaches are usually unable to identify such large-scale number of at-risk students in danger of dropping out in time to support effective intervention design. While building dropout prediction models using learning analytics are promising in informing intervention design for these at-risk students, results of the current prediction model construction methods do not enable personalized intervention for these students. In this study, we take an initial step to optimize the dropout prediction model performance toward intervention personalization for at-risk students in MOOCs. Specifically, based on a temporal prediction mechanism, this study proposes to use the deep learning algorithm to construct the dropout prediction model and further produce the predicted individual student dropout probability. By taking advantage of the power of deep learning, this approach not only constructs more accurate dropout prediction models compared with baseline algorithms but also comes up with an approach to personalize and prioritize intervention for at-risk students in MOOCs through using individual drop out probabilities. The findings from this study and implications are then discussed.

¹Department of Learning and Performance Systems, Pennsylvania State University, University Park, PA, USA

²Department of Industrial Engineering, Texas Tech University, Lubbock, TX, USA

Corresponding Author:

Wanli Xing, Department of Learning and Performance Systems, Pennsylvania State University, University Park, PA, USA.

Email: wanli.xing@ttu.edu

Keywords

MOOCs, dropout prediction, deep learning, intervention, personalization, machine learning

Introduction

As an extension of online learning technologies, the rapid development of MOOCs has opened a new era of education by extending the boundaries of education to previously noncollege bound students through the Internet (Fei & Yeung, 2015; Author 1). [AQ1] Since MOOC's first appearance in 2008, coined by Downes and Siemens, the number of MOOCs has grown enormously around the world. The total number of MOOCs reached 4,550 in 2016 (Bouzayane & Saad, 2017a). Due to its online and open nature, a MOOC course is usually massive, with theoretically no limit to enrollment and allows anyone to participate or drop out at no penalty (Educause, 2013, p. 1). As a result, students enrolled in MOOCs, unlike campus-confined students, are much more likely to drop out from the course, which can easily reach 90% attrition rates (Li et al., 2016; Taylor, Veeramachaneni, & O'Reilly, 2014). While the high incompleteness rate is often cited as a scale-efficacy tradeoff (Onah, Sinclair, & Boyatt, 2014), it does put a major obstacle to the transformative potential of MOOCs.

The substantial number of students enrolling in and dropping out of the MOOCs raise methodological difficulties for instructors as they work to identify academically at-risk students and provide in-time intervention. It is difficult, if not impossible, to monitor more than 10,000 students at the same time and then readily offer support to about 9,000 students who are about to dropout. The situation becomes worse considering traditional educational researchers and practitioners have been employing methods such as surveys, interviews, and observations for data collection. Such methods are not only time consuming but also limited in scale to identify potentially at-risk students in MOOCs (Author 2) [AQ2]. These traditional methods are unable to support instructors to provide timely interventions for at-risk students either (Rienties, Cross, & Zdrahal, 2017; Siemens & Long, 2011), since a delay always occurs between data collection, analysis, and finally identification of students who are about to dropout from the course.

The excessive attrition rate in MOOCs has encouraged researchers to think of using learning analytics methods (Siemens, 2013) for the early prediction of learners who are at risk of dropping out. Then, appropriate intervention can be delivered before the student drops out. Specifically, learning analytics enables analysis of low-level trace or log data automatically collected while students are interacting with the MOOC course. Then, through this structured low-level trace

data, prediction models can be constructed using supervised machine learning algorithms (Kennedy, Coffrin, De Barba, & Corrin, 2015), which leads to training a mathematical model to automatically make an early distinction between dropout and retention students based on their prior interaction behavior during the course. After making the distinction, in-time intervention can be designed and delivered to these at-risk students. The automatic nature of a prediction model using learning analytics can address the challenge of monitoring and identifying the large-scale at-risk students of potentially dropping out, while also satisfying the requirement for being able to support early intervention design (Halawa, Greene, & Mitchell, 2014).

Current research and practices in building prediction models in MOOCs follow three interrelated directions: fixed term dropout prediction model, temporal dropout prediction model, and dropout prediction performance optimization. These methods have serious limitations for deployment in MOOCs overall. Fixed-term dropout prediction model can identify all the at-risk students at once. However, given the large number of dropout students in MOOCs, the instructors cannot provide effective feedback to that many at-risk students at once. The temporal dropout prediction model predicts students at risk of dropping out in the next week. While it tremendously reduces the number of at-risk students the instructor needs to deal with every time, it does not provide any clue for the instructor to offer personalized intervention. The result is that all the students receive the same intervention without any personalization. Much effort on optimizing prediction performance has also been made. However, no studies so far take the advantage of deep learning which has shown great success in so many disciplines (Gulshan et al., 2016). In the next three sections, empirical studies for prediction model building are reviewed in more detail based on the proposed directions.

Fixed-Term Dropout Prediction Model

Many early efforts in building prediction models employed the fixed-term dropout prediction approach. That is, they used the data available in a fixed period of time for prediction model construction. Some studies used all the data in a course to build the prediction model, which could not satisfy the early intervention design and implementation requirement. For example, Al-Shabandar, Hussain, Laws, Keight, and Lunn (2017) used all-time trace data from a course to build the dropout prediction model. However, such a model is unable to identify the at-risk students early enough to support intervention design and delivery before the student drops out.

Other studies used the trace data only from the first week or certain points of time to build the prediction models (e.g., Jiang, Williams, Schenke, Warschauer, & O'dowd, 2014). This prediction building method can successfully detect whether students are at risk of dropping out in the very early stage (as early

as the end of the first week) to support early intervention design and implementation. However, this built model is unable to accommodate the gradual drop out pattern in MOOCs. In other words, this fixed-term model cannot detect who are at risk of dropping out after only the first week or which ones will remain active after 2, 3, or 4 weeks before dropping out. Consequentially, thousands of students may be flagged as being in danger of dropping out after the first week. While effective at predicting how many students will eventually dropout of a course, it is impractical for the instructors to provide quality help and support to thousands of at-risk students at the same time. In other words, this model is unable to identify those students in need of immediate intervention. Neither can it support personalized intervention for these at-risk students.

Temporal Dropout Prediction Model

Because of the gradual nature of attrition in MOOCs (Yang, Sinha, Admson, & Rose, 2013), recent studies began to explore building temporal prediction dropout models to accommodate better intervention design in MOOCs (Balakrishnan & Coetzee, 2013; Bouzayane, & Saad, 2017a, 2017b; Boyer & Veeramachaneni, 2015; Chaplot, Rhim, & Kim, 2015; Crossley, Paquette, Dascalu, McNamara, & Baker, 2016; Fei & Yeung, 2015; He, Bailey, Rubinstein, & Zhang, 2015; Kloft, Stiehler, Zheng, & Pinkwart, 2014; Li et al., 2016). To illustrate, these studies design a temporal modeling mechanism using the trace data until the current week, through which they can predict who is going to dropout next week. More specifically, instead of using the fixed-term data to identify all the students at risk at once, these temporal dropout prediction models allow specifically detection of at-risk students in the following week using data collected from previous weeks. Through only calling attention to those at-risk students in the coming week, this temporal prediction mechanism allows instructors to focus on a much smaller group of at-risk students in immediate danger instead of facing an overwhelming number of students who may drop out at some other points in the course.

However, these temporal models were not able to personalize student intervention. Personalization is one of the fundamental functions in learning analytics research and applications (Papamitsiou & Economides, 2014; Siemens & Baker, 2012). It also responds to the general call from the U.S. Department of Education (2012) for personalizing students' learning experience to improve their achievement through learning analytics. Currently, most dropout prediction models can only identify a large group of students at risk of dropping out but are unable to further provide concrete advice on how to personalize the intervention. In addition, these temporal dropout prediction models can only zoom into a group of at-risk students one week at a time. However, even for a group of students who have the risk of dropping out in a week, the number of students identified can still be large in MOOC context. Then, the MOOC system

becomes overwhelmed again with the unfulfilled need for the instructors to provide intervention.

To support personalized intervention design and delivery, one way is to compute the dropout probability of individual students each week. Instructors can then provide intervention based on the dropout probability of student attrition by providing more intensive and heavier intervention for the higher risk students, and lighter interventions for the lower risk students. In addition, students can also be ranked based on dropout probability. Then instructors are asked to provide immediate intervention to students who have a higher probability of attrition first and then gradually move to the lower risk students. This ranking can help teachers personalize intervention to the students who need the most help and immediate assistance.

Dropout Prediction Model Performance Optimization

Quite a few studies have examined different features and algorithms in order to improve prediction model performance in MOOCs (Crossley et al., 2016; Liang, Li, & Zhang, 2016; Qiu et al., 2016; Robinson, Yeomans, Reich, Hulleman, & Gehlbach, 2016). [AQ3] For example, Dmoshinskaia (2016) incorporated sentiment features from user comments into the prediction model, thereby improving the prediction accuracy. Ye and Biswas (2014) showed that finer-grained temporal information can also improve dropout prediction performance. From algorithmic perspective, Bouzayane and Saad (2017a) proposed a dominance-based rough set approach to improve the dropout prediction performance in MOOCs. Cobos, Wilde, and Zaluska (2017) compared several algorithms including generalized boosted regression models, weighted k-Nearest neighbors, boosted logistic regression, and gradient boosting for dropout prediction model building and found that generalized boosted regression model has the best prediction performance.

With the great advancement in computing power, the deep learning algorithm has shown great success in various fields (Deng & Yu, 2014; LeCun, Bengio, & Hinton, 2015). However, no studies have been found to explore the potential of deep learning algorithms in the education area, particularly the MOOCs context for dropout prediction. In fact, in light of the data variability and massiveness in MOOCs, as well as the highly imbalanced nature of dropping out over retention (Fei & Yeung, 2015), the deep learning algorithm has the potential to achieve better prediction performance compared with traditional algorithms. Such better performance can be reflected in both the prediction accuracy as well as the performance in producing individual dropout probabilities for intervention personalization.

Summarization and Research Goals

Building dropout prediction models using learning analytics has shown tremendous potential in informing intervention design to reduce the attrition rates in

MOOCs. However, current dropout prediction models are limited by either their support of early intervention design (fixed-term prediction dropout model) or not being able to prioritize individual dropout probability for personalized intervention (temporal dropout prediction model) in MOOCs. In addition, none of these previous studies have taken the advantage of the power of the deep learning algorithm in building dropout prediction models for model performance optimization.

Accordingly, two research goals guide this research: (a) to explore the power of deep learning technique in building weekly temporal dropout prediction models so we can determine if it can outperform the traditional-used machine learning approaches; (b) to investigate personalized intervention using individual dropout probabilities and further examine whether the deep learning algorithm can better personalize and prioritize intervention for students who are at risk of dropping out in MOOCs than other algorithms.

Methodology

Research Context and Data

The context for this research is a project management MOOC course hosted by Canvas. This course started in August 2014 and lasted roughly 8 weeks. In total, there were 11 modules and 3,617 students registered in this course. Also, 14 discussion forums and 12 multiple choice quizzes were structured in this MOOC course. The data set used in this study came from two sources: (a) all the click-stream or trace data provided directly by Canvas containing information on pages visited including when and how many students clicked on certain sources (e.g., syllabus, announcements, quizzes, assignments, submissions, etc.) and (b) discussion forum data and quiz scores retrieved in JSON format using the Canvas API. This data set includes quiz scores for every student and all the discussion content as well. Figure 1(a) depicts an overview of the course and Figure 1(b) shows how many students were active each week.

Features

Based on the review of the features used in previous studies, we found that different studies often used different features (Crossley et al., 2016; Fei & Yeung, 2015). However, they all correspond to the same set of underlying characteristics, which are the raw behavioral features that most MOOC platforms keep track of. In this current study, since feature engineering is not the main goal, we used a flat feature structure as most studies used in previous research as shown in Table 1. In addition, since this study also plans to build temporal dropout prediction models on a weekly basis, the features created in the

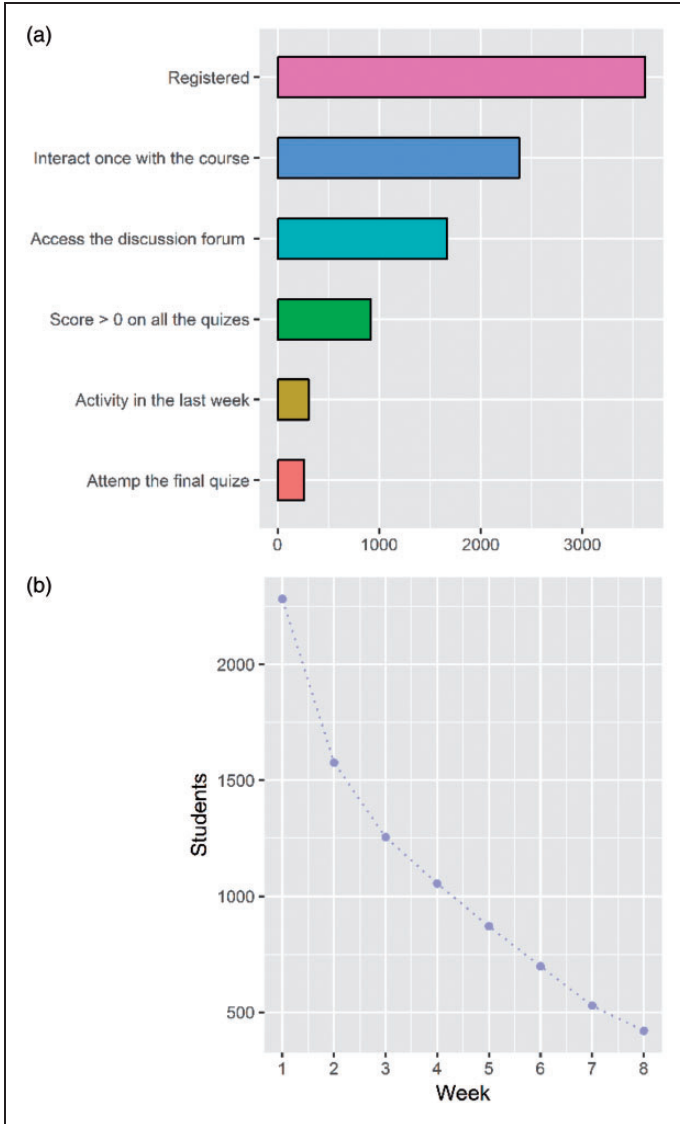


Figure 1. Course characteristics: (a) Overview and (b) active students [AQ9].

second week were directly added to the features created in the first week, and similarly for other weeks as well. Overall, we will use the features until the current week to predict whether a student is about to dropout in the next week.

Table 1. Features and Description.

Feature	Description
Number of announcements	Number of times students view the announcements
Number of assignments	Number of times students access the assignments
Number of calendar	Number of times students view the calendar
Number of module pages	Number of times students access the module pages
Number of courses	Number of times students access the courses
Number of discussion	Number of times students access the discussion forum
Number of files	Number of times students access the files
Number of gradebooks	Number of times students check the gradebooks
Number quizzes	Number of times students access the quizzes
Number of submissions	Number of times students submit assignments
Number of Wikis	Number of times students access the wikis pages
Number of actives days	Number of days student interacts with the course
Dropout week	The week when the student last visits the course. This is for algorithms to predict.

Algorithms and Evaluation

In this study, three popularly used algorithms were implemented as baseline algorithms: K-nearest neighbors (KNN), support vector machines (SVM), and decision tree. Then, the deep learning algorithm was developed to compare the prediction performance with the baseline algorithms.

The KNN algorithm is one of the most common classification methods. It has little or no prior requirement about the distribution of the data. KNN was originally derived from the necessity to conduct discriminant analysis when reliable parametric estimation of probability densities is difficult to determine (Denoeux, 1995). KNN has a very simple working mechanism in which each input pattern to be classified is compared with a series of stored patterns, in our case, dropout or not. The classification result is the class with the most representatives in the k retrieved patterns. The stored pattern in this study is the training data set, and the k is estimated using the hyperparameter optimization algorithm (Thornton, Hutter, Hoos, & Leyton-Brown, 2013). In Figure 2, we only use two features or dimensions to illustrate the algorithmic concept, which is easy to explain. KNN can handle as many dimensions as needed.

Decision tree is a typical top-down tree growth algorithm (Quinlan, 2014). This algorithm divides a complex classification task to a set of simple classification tasks. The decision tree is usually made of a root node, a group of internal nodes, and leaf nodes as shown in Figure 3. The root nodes refer to the features

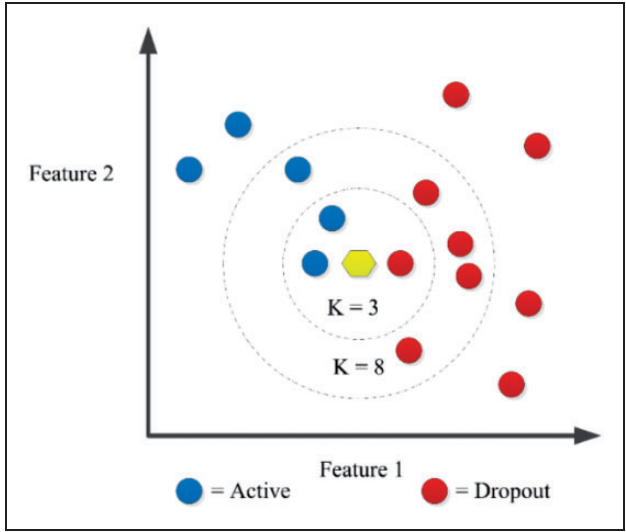


Figure 2. KNN with two-dimensional feature space and two classes: When K is set to three, then the hexagon is classified into the active group. If K is set to eight, then the hexagon is classified into the dropout group.

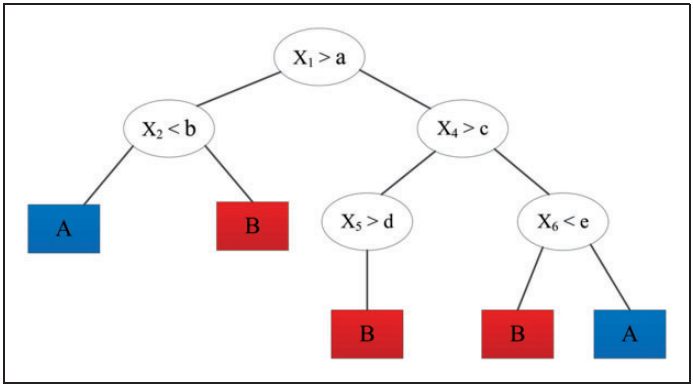


Figure 3. Decision tree with six-dimensional feature space and two classes: The X_i are the features and $a, b, c, d,$ and e are the thresholds and A, B are the class labels as active and dropout.

used to divide the samples. Each root node can produce a branch. Each path from the root node to the leaf node in the tree can be viewed as a classification rule. The decision tree algorithm must select the node feature value. In this study, the selection method mainly used the information gain ratio of splitting.

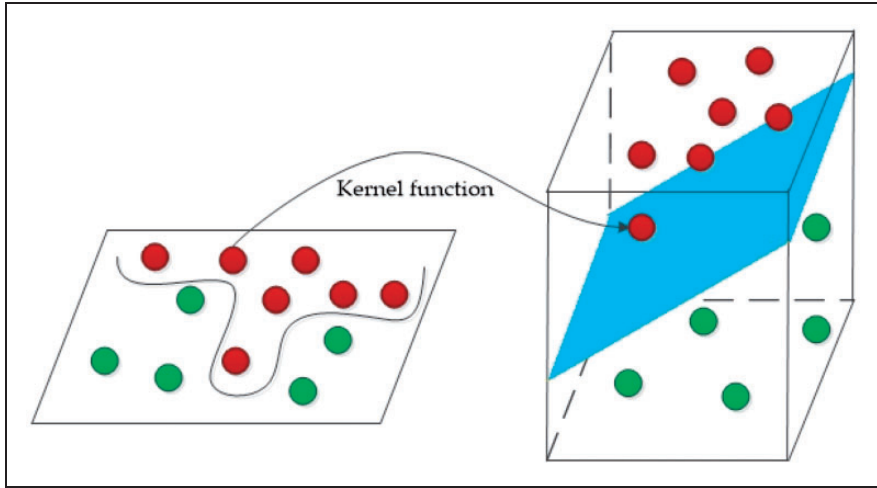


Figure 4. SVM schematic diagram.

All students will be classified as potentially at risk or active using the rules in the tree structure as shown in Figure 3.

Based on sound statistical learning theory, the performance of SVM is robust to data noise (Widodo & Yang, 2007). SVM is originally designed for binary classification by constructing an optimal separating hyperplane that has high generalization ability. Let $(x_1, y_1), \dots, (x_m, y_m) \in R^N \times (\pm 1)$ be the training data which are mapped by $\Phi(x)$ into a high dimensional feature space (Figure 4). Then, the linear hyperplane function can be constructed as $f(\Phi(x)) = w \times \Phi(x) + bw \in R^N, b \in R$.

In feature space, the optimal hyperplane can be expressed as the linear combination of training samples. These samples, as support vectors, construct the decision functions using the kernel function. Any function that meets Mercer's theorem can be used as a kernel function, which must be continuous and positive definite (Cristianini & Shawe-Taylor, 2000). Such kernel function could be linear, quadratic, polynomial, and Gaussian. Gaussian Radial Basis function has been demonstrated as one of the most efficient kernel functions (Scholkopf et al., 1997). Therefore, we chose to use the Gaussian Radial Basis function to define the feature space.

Deep learning is a format of neural network that takes metadata as input and then processes the data through a number of layers to compute the output (LeCun et al., 2015). While traditional neural network can only handle single hidden layer (Figure 5, left), deep learning processes the input data through a large number of hidden layers in its structure (Figure 5, right). Each layer is made of nodes, which is the place for computation to take place. A node combines input from the data with a set of weights to determine whether to amplify

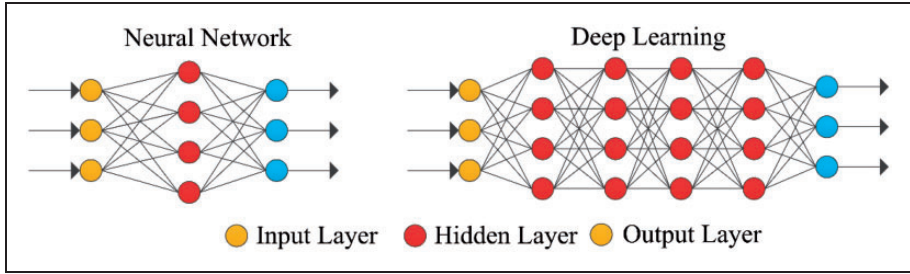


Figure 5. Deep learning diagram.

or dampen the input, which in turn assigns significance to the inputs. These input weight products are then summed and evaluated to decide to what extent the information propagates through the network to finally influence the classification. In a more holistic view, the hidden layer trains the unique set of features using the output of the previous layer. This process is known as non-linear transformation. The more hidden layers it has, the more complex and abstract the data will become.

In addition, the traditional neural network also requires more information about features for conducting the feature selection and feature engineering. By contrast, the deep learning neural network has no requirement for any information about features (Schmidhuber, 2015). It performs automatic feature extraction without any human intervention, grasping the relevant features necessary to solve the problem. In other words, deep learning performs optimum model tuning and selection on its own, which saves a lot of human effort and time. In this study, a customized neural network was designed for each individual week; in addition, forward and backward propagation algorithms were used to train the weights, and batch gradient descent was adopted to minimize the cost function.

To better evaluate the prediction performance of these algorithms, the data were divided into 70% for training and 30% for testing. To avoid overfitting, 10-fold cross-validation was performed, where performances were measured from multiple rounds of cross-validation and averaged over the rounds. The area under the receiver operating characteristic curve (AUC) was obtained to measure the prediction performance. Traditional precision, recall, and F-measure are usually valid only for one specific operating point, which is selected to minimize the probability (Bradley, 1997). However, selecting only one point can generate ambiguous results when comparing systems (Hand, 2009). By contrast, AUC is invariant to the selected decision criterion. Empirical results found that AUC can decrease the standard error compared with traditional metrics (Bradley, 1997). In addition to AUC statistics, accuracy was also calculated to show the robustness of the proposed deep learning algorithm.

Intervention Personalization

After using dropout prediction models to identify the MOOC students most likely to dropout each week, further analysis can be conducted to find ways to personalize the intervention. This study proposes to produce the dropout probability of each student in each week. Then, instructors and intelligent agents can use the probability of dropping out to personalize and prioritize intervention for at-risk students. However, not all the algorithms have the capability to produce the dropout probability for each student. In our study, while SVM, decision tree, and deep learning can generate the probability of dropping out for each individual student, KNN is not able to produce such probability.

To validate our intervention personalization proposal, regression analysis was used to examine whether the calculated individual students' dropout probabilities were correlated with the actual dropout date of the students for SVM, decision tree, and a deep learning algorithm. In these tests, dropout probability served as the independent variable X , and the specific dropout date of each student in a week was the dependent variable Y . If the coefficient of the independent variable X is significant, we could make the reasonable assumption that the classification model provided a good dropout rate prediction, which can be further used for personalized intervention. The rationale behind this is that the higher dropout probability indicates that the student is more likely to drop out early in the week. Then, it makes sense to provide stronger and prioritized intervention to these students as a way of personalization. By comparing the regression tests between algorithms, it can also demonstrate which algorithm generating probability would be more in line with individual student attrition date, which, in turn, demonstrates the advantage and validity of personalizing and prioritizing individual student intervention using the deep learning algorithm.


Results

Prediction Performance

Table 2 and Figure 6 present the predictive modeling performance using training data over the duration of the course. The range of AUC for KNN was 0.863 to 0.947 and for the decision tree the range was from 0.813 to 0.888. The performance of SVM ranged from 0.855 to 0.976. When it comes to deep learning algorithm, the range was from 0.954 to 0.984. The results from the AUC values show that the lowest prediction performance bound of the deep learning algorithm still outperformed the highest bound of the KNN and decision tree algorithms. This trend can also be observed in Figure 6(a). In terms of SVM, deep learning also performs better than the prediction performance of SVM for all the weeks. When averaging the whole MOOC course, KNN can reach 0.897 for the AUC and the decision tree can achieve 0.849. SVM can reach 0.909.

Table 2. AUC Results.

		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
KNN	AUC	0.886	0.863	0.896	0.892	0.901	0.892	0.947
	Accuracy	0.929	0.926	0.932	0.946	0.953	0.969	0.966
SVM	AUC	0.915	0.902	0.910	0.855	0.894	0.911	0.976
	Accuracy	0.988	0.892	0.911	0.914	0.941	0.966	0.944
DT	AUC	0.881	0.854	0.888	0.836	0.797	0.813	0.876
	Accuracy	0.955	0.953	0.960	0.964	0.971	0.984	0.982
DL	AUC	0.965	0.955	0.954	0.960	0.973	0.983	0.984
	Accuracy	0.928	0.940	0.958	0.966	0.957	0.981	0.974

Note. AUC = area under the receiver operating characteristic curve; KNN = K-nearest neighbors; SVM = support vector machines; DT = decision tree; DL =  [AQ8].

Deep learning has the highest average in AUC values as 0.968. Deep learning outperforms the baseline algorithms and ranges from 0.06 to 0.118. In fact, in this study, mostly three layers were used in building the deep learning algorithm. Sufficient room remains to further refine the performance of our deep learning algorithm.

When it comes to dropout prediction accuracy, the range for KNN was 0.926 to 0.966 according to Table 2 and Figure 6(b). The accuracy range for SVM was 0.892 to 0.988. The range for decision tree was 0.953 to 0.984. For deep learning algorithm prediction accuracy, the range was from 0.928 to 0.981. The accuracy of deep learning still has a better performance than the SVM algorithm (a 0.958 in average vs. 0.937). From the face values, it seems that KNN is very close to the prediction accuracy performance of deep learning (0.946 in average vs. 0.958 in average), and the accuracy of decision tree even outperforms the deep learning algorithm (0.967 in average vs. 0.958 in average). However, when looking at the performance of the trained predictive models on the test data set as shown in Table 3 and Figure 7, obviously, some serious overfitting problem for KNN and for the decision tree occurred in the training data set.

Specifically, for the prediction performance in the testing data set, the accuracy for KNN ranged from 0.865 to 0.947 with an average of 0.904. The accuracy for the decision tree was from 0.880 to 0.95 and the mean value was 0.915. By comparison, deep learning had a much more stable and higher prediction accuracy ranging from 0.905 to 0.961 with an average of 0.930. SVM was relatively stable from training to testing with a range from 0.869 to 0.953. However, the predictive performance was not as good as the deep learning algorithm. It is generally agreed in the research community that prediction performance in the testing data set is usually closer to the actual performance in real context (Lu, Kolarik, & Lu, 2011). After all, testing data were the data which were intangible to the machine learning algorithms. Overall, the deep learning algorithm had a

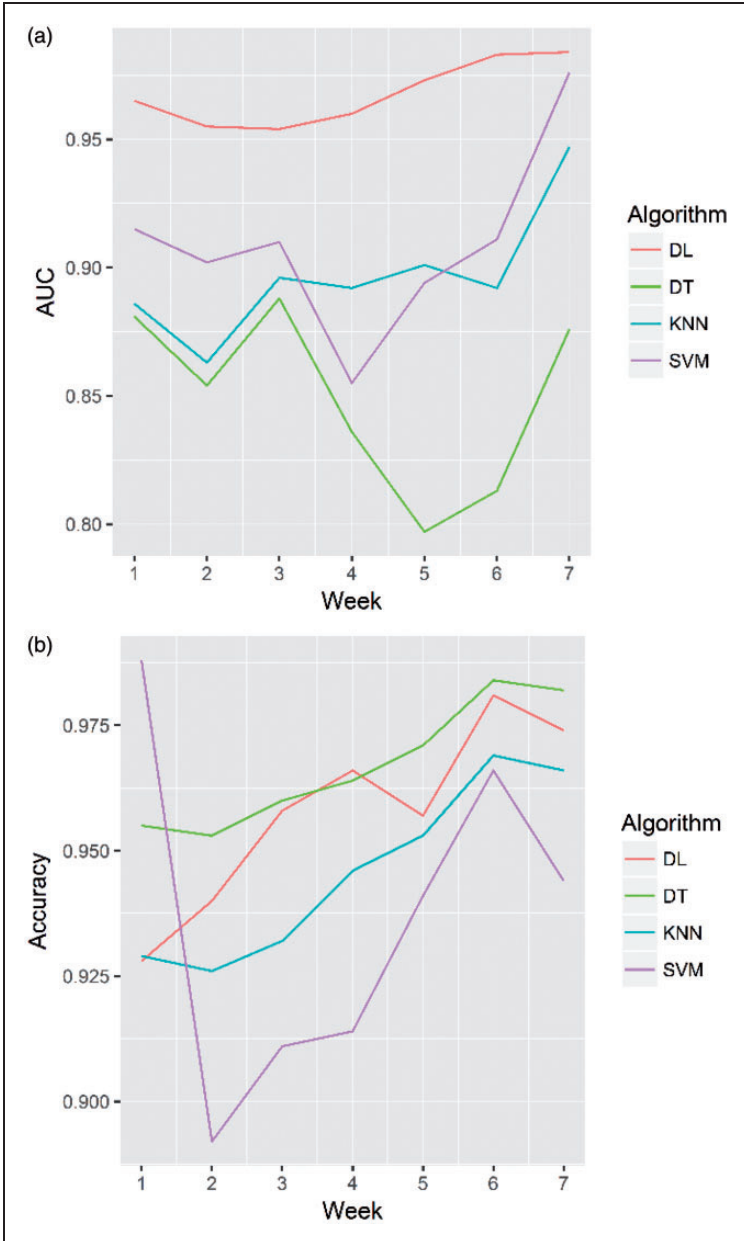


Figure 6. Prediction performance for training data: (a) AUC and (b) accuracy.

Table 3. Prediction Accuracy Over Testing.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
KNN	0.865	0.88	0.896	0.88	0.918	0.945	0.947
SVM	0.889	0.869	0.907	0.908	0.938	0.953	0.945
DT	0.88	0.887	0.917	0.915	0.914	0.95	0.944
DL	0.908	0.905	0.915	0.917	0.946	0.961	0.961

Note. KNN = K-nearest neighbors; SVM = support vector machines; DT = decision tree.

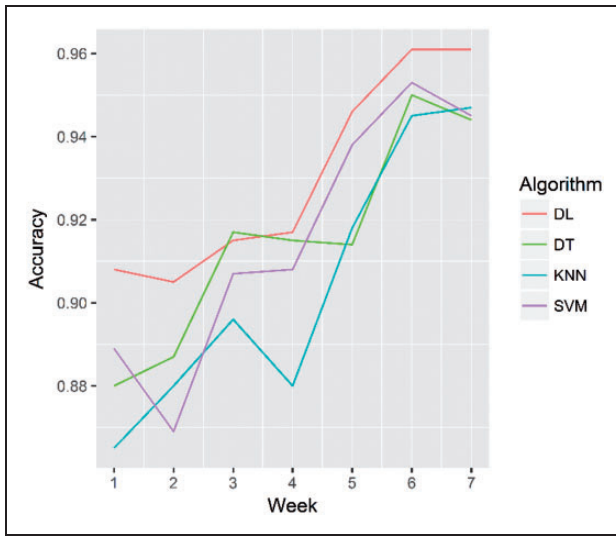


Figure 7. Testing prediction performance.

much better and more stable performance in terms of accurately identifying the at-risk dropout students in MOOCs.

Dropout Probability for Personalization

This study went a step further to propose using the predicted dropout probability to personalize and prioritize intervention. Since not all the algorithms (e.g., KNN) were able to generate such dropout probability, this study focused on the remaining three algorithms: SVM, decision tree, and the deep learning algorithm. As shown in Table 4, each of these three algorithms produced the dropout probability for each individual student. Since this study relied on a weekly temporal dropout model for each student dropout probability, a predicted dropout

Table 4. Dropout Probability for Sample Students.

	Support vector machines		Decision tree		Deep learning		Actual date
	Probability	Date	Probability	Date	Probability	Date	
Student 1	.511	7	.600	6	.310	3	3
Student 2	.885	2	1.000	1	.313	4	2
Student 3	.932	1	.571	7	.310	3	5
Student 4	.822	3	1.000	1	.318	5	2
Student 5	.646	5	.750	5	.310	3	6
Student 6	.939	1	.778	4	.124	2	2
Student 7	.948	1	.998	3	.310	3	4
Student 8	.939	1	.778	4	.124	2	2
Student 9	.824	3	1.000	1	.311	4	4

date was given in relation to Days 1 to 7 in the particular study week. From this format, two personalizations were accomplished: (a) based on the individual dropout probability, instructors of the MOOC course provided more intensive interventions to the students with a higher probability of dropping out. Intelligent feedback was also designed using a probability distribution over the students to provide more personalized feedback. (b) Instructors also used this dropout probability ranking to prioritize the intervention, giving more intervention and help to the students who had the most immediate needs related to their dropout date.

To be more specific about the personalized intervention mechanism, we take the SVM predicted results as an example shown in Table 4. Using the log data until the current week, SVM can identify all the students at risk of dropping out for next week along with their dropout probabilities. Then for Students 1 and 6, the MOOC instructor can pay more attention to Student 6 first since the dropout probability for this student is higher than the Student 1 ($0.939 > 0.511$). Also, given the dropout probability for Student 6 is higher, the instructor can also provide heavier intervention to this student. While only sending one e-mail to Student 1, the instructor can send several e-mails with different wording to the Student 6. In reality, the instructor can also place the at-risk students into several groups based on the rank of the dropout probability. Then, a specific composed e-mail with different emphasis can be sent to each group or dealing with each group with different intervention strategies for personalized help.

To test the validity of our personalized intervention proposal, linear regression tests for each algorithm were conducted between the individual predicted probability and the actual dropout date as shown in Figure 8. The results reflect that deep learning has the best trend alignment between the dropout probability

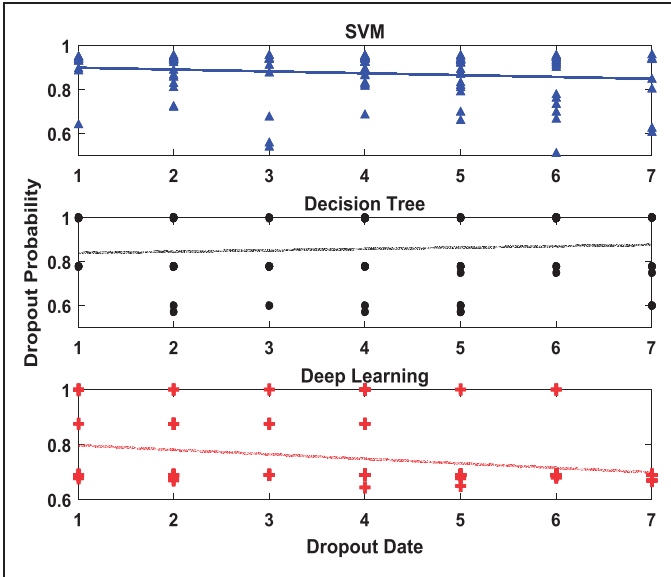


Figure 8. Regression tests for personalization.

and the actual dropout date ($p = .000$, adjusted $R^2 = 0.0687$). The decision tree had the least alignment between these two variables ($p = .320$, adjusted $R^2 = -4.09e-05$), and SVM fell in between deep learning and decision tree ($p = .0341$, adjusted $R^2 = 0.0162$). The higher adjusted R^2 value and the small p value of the deep learning model indicate larger correlations between probabilistic dropout predictions and dropout time. These significant tests show that it makes sense at least from statistical perspective to prioritize and personalize heavier intervention to students with higher predicted dropout probability since they are more likely to dropout early. In addition, the more aligned and significant trend in the deep learning regression test indicated that deep learning also has an advantage in personalizing and prioritizing interventions for at-risk students than the other algorithms.

Discussion

MOOCs have become more and more popular because they show great potential to transform the traditional education system (Bouzayane & Saad, 2017; Kloft et al., 2014) [AQ4]. However, due to their unique characteristics of being fully open and online, students dropout literally at every point of the course (Yang et al., 2013). As thousands of students keep dropping out throughout the course, it raises methodological difficulties for researchers and instructors to employ traditional educational approaches (e.g., questionnaires, interviews,

and observations) to inform intervention design and delivery for the at-risk students of dropping out. Traditional methods are very time consuming and unable to identify such a large-scale number of at-risk students in time.

Researchers are beginning to explore the use of learning analytics to build dropout prediction models for early identification of at-risk students and then deliver interventions before the student actually drops out. However, the current dropout prediction model building has significant restraints in dealing with the massive nature of MOOCs and at the same time personalizes intervention design. For example, many studies used fixed-term dropout prediction model, which is to build a dropout prediction model using data from a certain time period (Al-Shabandar et al., 2017; Jiang et al., 2014). This prediction model identifies all the at-risk students in a MOOC course at once, which makes it impossible for instructors to design interventions for so many at-risk students at the same time. While the new trend of using weekly temporal dropout prediction models is becoming more dominant in the research community (Balakrishnan & Coetzee, 2013; Bouzayane, & Saad, 2017b; Crossley et al., 2016), simply identifying a group of students who are in danger of dropping out each week still has significant limitations in terms of personalizing intervention (Author 1). That is, the number of students identified each week can still be large and instructors have no information using these prediction models to personalize their intervention design and delivery.

Based on the limitations in the current dropout prediction model construction in MOOCs, this study proposed an approach, while still relying on weekly temporal dropout prediction models, to further produce all the student predicted dropout probabilities as a way to personalize the intervention design and delivery. Given the generated student dropout probability, instructors of MOOCs can easily decide to provide more intervention and support for students who have higher probabilities of dropping out. Meanwhile, given the ranking of the probabilities of dropping out, instructors can also prioritize their intervention design so that they have an even smaller number of at-risk students to deal with each day. Otherwise, instructors may also have to deal with all the at-risk students each week at once, which would be challenging given the possible high number of at-risk students identified (Kloft et al., 2014). This personalization proposal is validated using regression techniques to correlate the student dropout probability with the actual dropout day in a particular week. The significant findings show that personalization can at least be statistically validated to personalize and prioritize intervention using the dropout probabilities.

In practice, the deep learning algorithm can be developed into a plugin for the MOOC platform. This plugin reads live log data from the database that collects all the students' digital traces while they are interacting with the MOOC courseware. Then, at the end of every week, the plugin can run once (run dynamically depending on the needs) and identify the at-risk students for the next week and generate their dropout probabilities at the same time. These at-risk students

along with the dropout probabilities can be visualized by the plugin and presented to the MOOC course instructor. Based on the visualization, the instructor can decide to provide intervention to certain students first and some students later in that week. Also, based on the dropout probabilities values, the instructor can provide heavier intervention for higher at-risk students and lighter for the lower risk ones.

In addition, the personalization and prioritization mechanism for intervention can also easily be implemented in an automated agent like in intelligent systems (Wenger, 2014). After all, instructors providing personalized help still have limitations given the time and effort required (Rienties et al., 2017). With an intelligent agent, it can directly deal with the dropout probabilities by considering its face value and rank. By comparing the dropout probability each student has with the thresholds set for different interventions, a predefined intervention strategy can be delivered for that at-risk student automatically. Such predefined intervention strategies can be learned from expert teachers or from historical data. It can be an automated e-mail, a reduced assignment, or even a video message delivered to the student. As said, this current study is just the first step toward the ultimate diverse and automated personalized intervention and scaffolding for students, and in turn to improve their engagement in MOOCs and their final success.

From an algorithmic perspective, this study explored the power of deep learning in education context. As deep learning has gained popularity and showed great effectiveness in various disciplines (LeCun et al., 2015), no studies were found to examine how deep learning can work in an educational context, particularly in an MOOC context. The amount of data generated by students in MOOCs is enormous with great variability (Fei & Yeung, 2015). The experiment in this study showed that deep learning performs consistently better than the other baseline algorithms including KNN, SVM, and decision tree. This superior performance from deep learning included not only the prediction accuracy in identifying at-risk students but also its capability in generating more accurate individual student dropout probabilities for intervention personalization and prioritization. In addition, by comparing the prediction performance of the different algorithms between training and test data, the results show that the prediction performance of deep learning is much more stable than the other algorithms as well. Also, we found that KNN and decision tree can become easily overfitted compared with SVM and deep learning algorithms by comparing their performance in dealing with training data and testing data. This finding has implications for the current research and practice in building prediction models in MOOCs as only reporting the dropout prediction performance in training data may not necessarily reflect the actual predictive power for MOOCs.

An important limitation for this study is that our findings are based on a single MOOC course data. Even though as a case study, the results supported the advantage of deep learning for prediction and personalized intervention design, the general application of the findings should be used with caution.

Another limitation is the proposed prediction and personalized intervention mechanism, although based on the literature support, is still hypothetical. It means that they have not been proved to be effective in live context. These are all sound directions for future research.

Conclusion

This study will optimize the dropout prediction models to focus on personalizing and prioritizing intervention for academically at-risk students in MOOCs. Relying on a weekly temporal prediction mechanism, this study proposes to use a deep learning algorithm to build dropout models and further produce individual student dropout probabilities for intervention personalization. By utilizing deep learning, this approach will not only build more accurate dropout prediction models compared with baseline models but also introduce a valid approach to inform intervention design thereby personalizing and prioritizing support for at-risk students using MOOC dropout probabilities.

Future research can go in three directions: (a) future studies can explore methods to further improve the deep learning prediction performance in MOOCs such as increasing the hidden layers in the network; (b) currently, the personalization and prioritization only show statistical validity. Researchers can further examine its validity by implementing the proposed prediction model and intervention personalization method in actual MOOC courses to assess if this approach can reduce students' dropout; (c) future research can conduct similar experiments on more general online courses to examine whether deep learning can be useful in other educational contexts.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Acknowledgment

We thank the Instructure Canvas and particularly Jared Stein for providing the data and support for this study.

References

- Al-Shabandar, R., Hussain, A., Laws, A., Keight, R., & Lunn, J. (2017). *Machine learning approaches to predict learning outcomes in massive open online courses*. Paper presented at the 2017 international joint conference on neural networks (IJCNN 2017), Anchorage, Alaska, USA.
- Balakrishnan, G., & Coetzee, D. (2013). Predicting student retention in massive open online courses using hidden markov models (Technical Report No. UCB/EECS-2013-109). Berkeley, CA: Electrical Engineering and Computer Sciences University of California at Berkeley.

- Bouzayane, S., & Saad, I. (2017a, May). Weekly predicting the at-risk MOOC learners using dominance-based rough set approach. *European Conference on Massive Open Online Courses* (pp. 160–169). Cham, Switzerland: Springer.
- Bouzayane, S., & Saad, I. (2017b, January). Prediction method based DRSA to improve the individual knowledge appropriation in a collaborative learning environment: Case of MOOCs. *Proceedings of the 50th Hawaii International Conference on System Sciences*. [AQ5].
- Boyer, S., & Veeramachaneni, K. (2015). Transfer learning for predictive models in massive open online courses. In Conati, C., Heffernan, N., Mitrovic, A. & Verdejo, M. (Eds.), *Artificial intelligence in education* (pp. 54–63). Cham, Switzerland: Springer June.
- Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), 1145–1159.
- Chaplot, D. S., Rhim, E., & Kim, J. (2015, June). Predicting student attrition in MOOCs using sentiment analysis and neural networks. *17th international conference on artificial intelligence in education*, Madrid, Spain.
- Cobos, R., Wilde, A., & Zaluska, E. (2017). Comparing attrition prediction in FutureLearn and edX MOOCs. *FutureLearn workshop in learning analytics and knowledge 2017* (pp. 1–20), Vancouver, Canada.
- Crossley, S., Paquette, L., Dascalu, M., McNamara, D. S., & Baker, R. S. (2016, April). Combining click-stream data with NLP tools to better understand MOOC completion. *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 6–14). Edinburg, England: ACM.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge, England: Cambridge University Press.
- Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4), 197–387.
- Denooux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5), 804–813.
- Educause. (2013). Seven things you should know about MOOCs II. *Educause Learning Initiative*. Retrieved from <http://net.educause.edu/ir/library/pdf/ELI7097.pdf>
- Fei, M., & Yeung, D. Y. (2015, November). Temporal models for predicting student dropout in massive open online courses. *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (pp. 256–263). Atlantic City, NJ: IEEE.
- Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., . . . Kim, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22), 2402–2410.
- Halawa, S., Greene, D., & Mitchell, J. (2014). Dropout prediction in MOOCs using learner activity features. *Experiences and Best Practices in and Around MOOCs*, 7, 3–12.
- Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1), 103–123.
- He, J., Bailey, J., Rubinstein, B. I., & Zhang, R. (2015, January). Identifying at-risk students in massive open online courses. *AAAI'15 proceedings of the 29th AAAI conference on artificial intelligence* (pp. 1749–1755). Austin, TX: AAAI Press.
- Jiang, S., Williams, A., Schenke, K., Warschauer, M., & O'dowd, D. (2014, July). Predicting MOOC performance with week 1 behavior. *Educational Data Mining 2014*. [AQ6].

- Kennedy, G., Coffrin, C., De Barba, P., & Corrin, L. (2015, March). Predicting success: How learners' prior knowledge, skills and activities predict MOOC performance. *Proceedings of the fifth international conference on learning analytics and knowledge* (pp. 136–140). New York, NY: ACM.
- Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014, October). Predicting MOOC dropout over weeks using machine learning methods. *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs* (pp. 60–65). Stroudsburg, PA: ACL.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Li, W., Gao, M., Li, H., Xiong, Q., Wen, J., & Wu, Z. (2016, July). Dropout prediction in MOOCs using behavior features and multi-view semi-supervised learning. *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 3130–3137). New York, NY: IEEE.
- Liang, J., Li, C., & Zheng, L. (2016, August). Machine learning application in MOOCs: Dropout prediction. *2016 11th International Conference on Computer Science & Education (ICCSE)* (pp. 52–57). New York, NY: IEEE.
- Lu, H., Kolarik, W. J., & Lu, S. S. (2001). Real-time performance reliability prediction. *IEEE Transactions on Reliability*, 50(4), 353–357.
- Onah, D. F., Sinclair, J., & Boyatt, R. (2014). Dropout rates of massive open online courses: Behavioural patterns. *6th International Conference on Education and New Learning Technologies, Barcelona, Spain* (pp. 5825–5834). **IAQT**.
- Papamitsiou, Z., & Economides, A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Journal of Educational Technology & Society*, 17(4), 49.
- Qiu, J., Tang, J., Liu, T. X., Gong, J., Zhang, C., Zhang, Q., & Xue, Y. (2016, February). Modeling and predicting learning behavior in MOOCs. *Proceedings of the ninth ACM international conference on web search and data mining* (pp. 93–102). New York, NY: ACM.
- Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. New York, NY: Elsevier.
- Rienties, B., Cross, S., & Zdrahal, Z. (2017). Implementing a learning analytics intervention and evaluation framework: What works?. In Kei Daniel, B. (Ed.), *Big data and learning analytics in higher education* (pp. 147–166). Cham, Switzerland: Springer International Publishing.
- Robinson, C., Yeomans, M., Reich, J., Hulleman, C., & Gehlbach, H. (2016, April). Forecasting student achievement in MOOCs with natural language processing. *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 383–387). New York, NY: ACM.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
- Scholkopf, B., Sung, K. K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., . . . Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11), 2758–2765.
- Siemens, G. (2013). Learning analytics: The emergence of a discipline. *American Behavioral Scientist*, 57(10), 1380–1400.
- Siemens, G., & Baker, R. S. (2012, April). Learning analytics and educational data mining: Towards communication and collaboration. *Proceedings of the 2nd*

- international conference on learning analytics and knowledge* (pp. 252–254). New York, NY: ACM.
- Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5), 30.
- Taylor, C., Veeramachaneni, K., & O'Reilly, U. M. (2014). Likely to stop? Predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382* [AQ10].
- Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013, August). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 847–855). New York, NY: ACM.
- U.S. Department of Education, Office of Educational Technology. (2012). *Enhancing teaching and learning through educational data mining and learning analytics: An issue brief*. Washington, DC: Author.
- Wenger, E. (2014). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. San Francisco, CA: Morgan Kaufmann.
- Widodo, A., & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6), 2560–2574.
- Yang, D., Sinha, T., Adamson, D., & Rosé, C. P. (2013, December). Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. *Proceedings of the 2013 NIPS Data-Driven Education Workshop*, 11, 14.
- Ye, C., & Biswas, G. (2014). Early prediction of student dropout and performance in MOOCs using higher granularity temporal information. *Journal of Learning Analytics*, 1(3), 169–172.

Author Biographies

Wanli Xing is an assistant professor in Instructional Technology at Texas Tech University, USA, with background in learning sciences, statistics, computer science, and mathematical modeling. His research interests are educational data mining, learning analytics, and CSCL.

Dongping Du is an assistant professor in Industrial Engineering at Texas Tech University, USA, with background in statistical modeling and management. Her research interests are big data analytics and computer simulation and optimization.