Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 8404653, 11 pages
https://doi.org/10.1155/2019/8404653

Research Article

MOOC Dropout Prediction Using a Hybrid Algorithm Based on Decision Tree and Extreme Learning Machine

Jing Chen,[1] **Jun Feng**,[1] **Xia Sun**,[1] **Nannan Wu**,[1] **Zhengzheng Yang**,[1] **and Sushing Chen**[2]

[1]*School of Information Science and Technology, Northwest University, Xi'an, China*
[2]*Computer Information Science and Engineering, University of Florida, Gainesville, FL, USA*

Correspondence should be addressed to Xia Sun; raindy@nwu.edu.cn

Received 1 November 2018; Revised 14 January 2019; Accepted 4 February 2019; Published 18 March 2019

Academic Editor: Eric Lefevre

Massive Open Online Courses (MOOCs) have boomed in recent years because learners can arrange learning at their own pace. High dropout rate is a universal but unsolved problem in MOOCs. Dropout prediction has received much attention recently. A previous study reported the problem of learning behavior discrepancy leading to a wide range of fluctuation of prediction results. Besides, previous methods require iterative training which is time intensive. To address these problems, we propose DT-ELM, a novel hybrid algorithm combining decision tree and extreme learning machine (ELM), which requires no iterative training. The decision tree selects features with good classification ability. Further, it determines enhanced weights of the selected features to strengthen their classification ability. To achieve accurate prediction results, we optimize ELM structure by mapping the decision tree to ELM based on the entropy theory. Experimental results on the benchmark KDD 2015 dataset demonstrate the effectiveness of DT-ELM, which is 12.78%, 22.19%, and 6.87% higher than baseline algorithms in terms of accuracy, AUC, and F1-score, respectively.

1. Introduction

MOOCs emerged as the natural solution to offer distance education with online learning enormously changing over the past years. MOOCs are widely used because of a potentially unlimited enrollment, nongeographical limitation, free accessibility for majority of courses, and structure resemblance with traditional lectures [1, 2]. Simply, they allow learners to learn anytime and anywhere at their own pace. With MOOCs booming popular [3, 4], the enrollment number of participants has increased from 8 million in 2013 to 101 million in 2018 rapidly [5, 6].

However, one critical problem that should not be neglected is that only an extremely low percentage of participants can complete courses [7–10]. Meanwhile, due to the high ratio of learner-to-instructor in online learning environment [8], it is unrealistic for instructors to track learners' learning behavior, which results in dropout or retention. Many educational institutions will benefit from accurate dropout prediction. That will help to improve the course design, content, and teaching quality [11–13]. On the other hand, it will also help instructors supply learners with effective interventions, such as proposing personalized recommendations of educational resources and guiding suggestions.

Dropout prediction has recently received much attention. Previous studies applied traditional machine learning algorithms to it. These algorithms include logistic regression [14–18], support vector machine [19], decision tree [20], boosted decision trees [2, 21, 22], and hidden Markov model [23]. However, there exists the problem of low accuracy leading to misidentification of at-risk learners, those who may quit courses.

Most recently, deep learning has become the state-of-the-art machine learning technique with a great potential for dropout prediction [24]. Jacob et al. applied a deep and fully connected feed forward neural network which is capitalized on nonlinear feature representations automatically. Fei et al. utilized a recurrent neural network model with long short-term memory (LSTM) cells which encoded features

into continuous states [25]. Although deep learning achieves more accuracy than traditional machine learning methods, it should be noted that deep neural networks need iterative training and a large amount of training data.

Moreover, due to the design discrepancy of MOOC platforms, current research utilizes different learning behaviors in dropout prediction [26]. Lack of uniform definition and understanding of learning behaviors in online learning environment [27, 28] will lead to un-unified conclusion on behavior features with better classification ability. The range of result fluctuation is widely resulting from the learning behavior discrepancy. Feature selection is essential in dropout prediction. Nevertheless, little attention has been devoted to it and most related studies utilize as many features as possible. Genetic algorithm is one of the common used feature selection methods with good scalability combining with other algorithms easily [29, 30]. However, it needs iterative training.

The goal of our approach is incorporating feature selection and fast training to realize accurate dropout prediction. To address feature selection, we adopt the decision tree algorithm due to its tree structure and theoretical basis. Further, the selected features are enhanced with different weights depending on the decision tree structure. The aim is to strengthen the features with good classification ability.

To realize fast training, we choose the ELM algorithm for dropout prediction. ELM is a single hidden layer feed forward neural network which improves the gradient algorithm and requires no updating parameters by repeated iterations [31, 32]. However, a theoretical guiding rule to determine the structure of ELM is lacking. Different structures lead to different prediction results.

To achieve accurate results of drop prediction, we map the decision tree structure to the ELM structure based on the entropy theory. The mapping rule takes full account of the impact of internal nodes on leaf nodes in the decision tree. It determines not only the neuron numbers of each layer in ELM, but also the connections between input layer and hidden layer. By this way, reasonable information assignment is realized at the initial stage of ELM.

In line with common practice in dropout prediction, we extract behavior features from raw learning records. Unlike past approaches, feature selection and enhancement are realized by decision tree. Then decision tree is incorporated with ELM to realize fast training and accurate prediction. Meanwhile, it is noteworthy that we utilize the same tree structure to solve the different problems. The core of our proposed algorithm is how to design the mapping rule to determine the structure of ELM.

The main contribution of this paper can be summarized as follows. Firstly, we define and extract several interpretive behavior features from raw learning behavior records. Secondly, we propose a novel hybrid algorithm combining decision tree and ELM for dropout prediction. It solves the problems of behavior discrepancy, iterative training, and structure initialization of ELM. It successfully makes full use of the same decision tree structure as a warm-start to the whole algorithm. Finally, we verify the effectiveness of our proposed algorithm by conducting experiments on the benchmark KDD 2015 dataset and it performs much better than baseline algorithms in multiple criteria.

2. Method

2.1. Problem Statement. There are three definitions of MOOC dropout prediction in the current studies. The first is whether a learner will still participate in the last week of the course [33–35]. The second is whether the current week is the last week a learner has activities [17, 19, 36]. Those two definitions are similar because they are related to the final state of a learner, and the dropout label cannot be determined until the end of the course. The third definition is whether a learner will still participate in the coming week of the course, which is related to the ongoing state of a learner [25, 37]. The dropout label can be determined based on the behavior of current week, which can help the instructors to make the interventions timely. Thus, the third definition is used in our paper.

The expectation confirmation model explains why users continue to use the information system [38], and then it is extended to explain why learners continue to use MOOCs [39]. The studies find that there exist several significant factors which can influence the continuing usage, such as confirmation of prior use, perceived usefulness, and learners' satisfaction. According to that, the current learning may have more impact on the intention of continuing usage. For most learners, if they confirm the usefulness and feel satisfied of current week learning, they may have strong intention to continue the learning in the next week.

Therefore, the goal of this paper is to predict who may stop learning in the coming week based on the learning behaviors of current week, which helps instructors better track the learning state of the leaner to take corresponding interventions. Assume there are r behavior features extracted from learning behavior records for the current week, which is represented as a r dimensional vector $x_i \in R^r$. $y_i \in R^m$ is the corresponding dropout label. If there are activities associated with i_{th} learner in the coming week, the dropout label of this learner is $y_i = 0$ which indicates that the learner will continue to learn. Otherwise, the dropout label is $y_i = 1$ which means the learner will quit the course next week.

2.2. Framework of MOOC Dropout Prediction. To address the problem of MOOC dropout prediction, we propose a framework which is shown in Figure 1. To be specific, the first module designs and extracts several features from learners' learning behavior records. The feature quantification is realized by calculating the number of each feature, which reflects the engagement of learners. The outputs of this module are feature matrix and label matrix.

The second module implements dropout prediction using DT-ELM algorithm based on the extracted behavior features. The decision layer is designed to select features and determine the ELM structure based on decision tree. It outputs the tree structure to the mapping layer and the selected features to the enhancement layer. The enhancement layer targets the strengthening of the classification ability of the selected features. It outputs the enhanced features to the improved

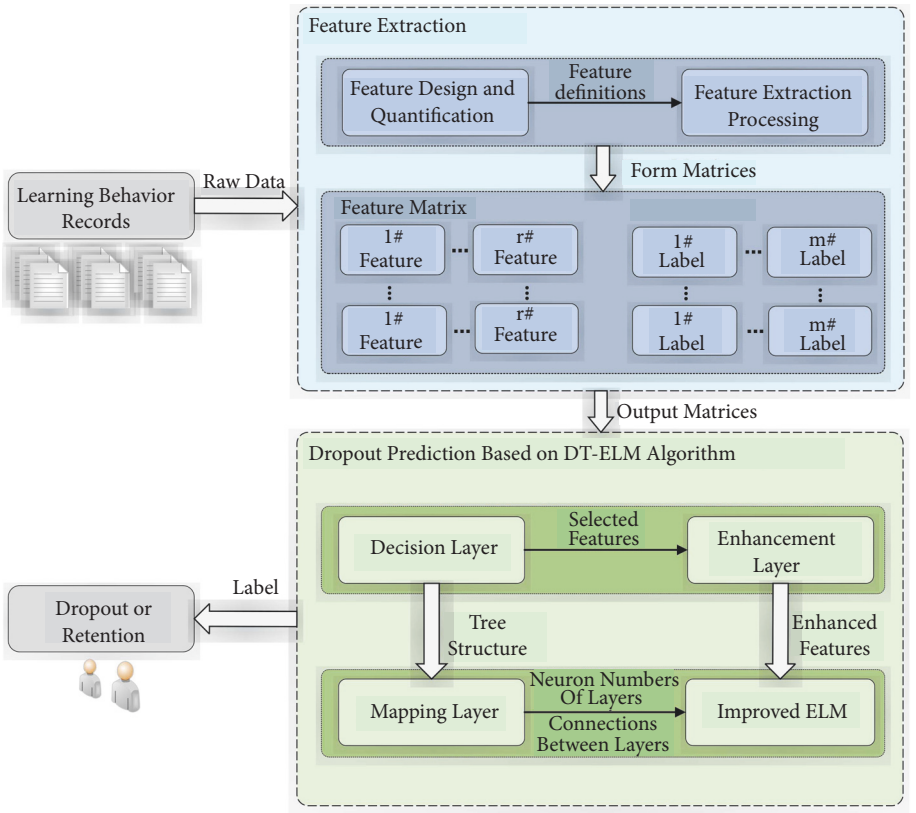

Figure 1: Overall framework of dropout prediction.

ELM. The function of mapping layer is to determine the ELM structure according to the tree structure. It outputs the neuron numbers of each layer and connections between layers. By the improved ELM, the dropout or retention can be obtained.

2.3. Feature Extraction. We extract features from learning behavior records. Courses generally launch weekly. It is better to utilize the numbers of learning behavior records by week as features [40]. The set for each type of learning behavior records is represented as R_t. Here t denotes the t_{th} type of learning behavior and $t = 1, 2, \cdots, r$. The record number of t_{th} type of learning behavior during the duration for each course is expressed as a vector $x_{it} = [x_{it}^{(1)}, x_{it}^{(2)}, \cdots, x_{it}^{(q)}, \cdots, x_{it}^{(W)}]$, where i represents the i_{th} leaner. $x_{it}^{(q)}$ is the number of learning behavior records in the q_{th} week, and W is the number of weeks a course lasts. The feature extraction process is shown in Algorithm 1. It outputs the feature matrix and label matrix.

After feature extraction, the feature matrix $x = [x_1; x_2; \cdots; x_e]$ is obtained, where e is the number of enrollment learners. $x_i = [x_{i1}, x_{i2}, \ldots, x_{ir}]$ represents the behavior features of the i_{th} learner. $y = [y_1; y_2; \cdots; y_e]$ is the label matrix, where $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}] \in R^m$ is the dropout label of the i_{th} learner.

Effective learning time is another kind of behavior feature and represents the actual time that a learner spends on learning. In practice, a learner may click a video and then leaves for something else. Therefore, we set a threshold between two activity clicks. The time exceeding the threshold will not be counted.

2.4. Dropout Prediction Based on DT-ELM Algorithm. Decision Layer. The decision layer implements the feature selection using decision tree based on the maximum information gain ratio [41]. D is the input of decision layer. Each instance in D is represented as $x_i = [x_{i1}, x_{i2}, \ldots, x_{ir}] \in R^r$. $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}] \in R^m$ is the class label of x_i. D' is the output of decision layer only containing the selected features. Each instance in D' is represented as $x_{i_{new}} = [x_{i1}, x_{i2}, \ldots, x_{in}]$ which means there are n selected features in D'.

Decision tree is constructed by recursive partitioning D into smaller subsets $D_1, D_2, \ldots, D_K$ until reaching the specified stopping criterion, for example, that all the subsets belong to a single class. A single feature split is recursively defined for all nodes of the tree using some criterion. Information gain ratio is one of the most widely used criteria for decision tree. The entropy which comes from information theory is described as

$$Info(D) = -\sum_{c=1}^{m} p_c \log_2(p_c) \quad (1)$$

where m represents the number of classes. p_c is the probability that an instance x_i belongs to the class c. The

Inputs:
R: Learning behavior records of a course
e: Enrollment number of learners
r: Number of behavior features
d: Duration of the course
Outputs:
x: Feature matrix with size of $e \times r$
y: Label matrix with size of $e \times m$
1: R is the set of learning behavior records for each course. It is grouped by the behavior types. Let $R = [R_1, R_2, \cdots, R_t, \cdots R_u]$ be the record set of learning behaviors, where $t = 1, 2, \cdots, r$.
2: Divide the duration of this course into W weeks.
3: For each learning behavior record in R_t
4: If this record occurred in week q generated by learner i
5: $x_{it}^{(q)} += 1$
6: For each learner, the t_{th} learning behavior feature $x_{it} = [x_{it}^{(1)}, x_{it}^{(2)}, \cdots, x_{it}^{(q)}, \cdots, x_{it}^{(W)}]$ is obtained.
7: Form the feature matrix $x = [x_1; x_2; \cdots; x_e]$, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{ir}]$ is r types of behavior features of the i_{th} learner.
8: Form the label matrix $y = [y_1; y_2; \cdots; y_e]$, where $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}] \in R^m$.

Algorithm 1: Feature Extraction Processing.

split rule is defined by information gain which represents the expected reduction in entropy after the split according to a given feature. The information gain is described as follows.

$$Gain(A) = Info(D) - Info_A(D) \tag{2}$$

$Info_A(D)$ is the conditional entropy which represents the entropy of D based on the partitioning by feature A. It is computed by the weighted average over all sets resulting from the split shown in (3), where $|D_k|/|D|$ acts as the weight of the k_{th} partition.

$$Info_A(D) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} \times Info(D_k) \tag{3}$$

The information gain ratio extends the information gain which applies a kind of normalization to information gain using a "split information" value.

$$SplitInfo_A(D) = -\sum_{k=1}^{K} \frac{|D_K|}{|D|} \times \log_2\left(\frac{|D_K|}{|D|}\right) \tag{4}$$

The feature with the maximum information gain ratio is selected as the splitting feature, which is defined as follows.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \tag{5}$$

The decision tree is constructed by this way. Each internal node of the decision tree corresponds to one of the selected features. Each terminal node represents the specific class of a categorical variable.

Enhancement Layer. Because the classification ability of each selected feature is different, the impact of this feature on each leaf node is different. The root node has the best classification ability and it connects to all leaf nodes. That means the root node has the greatest impact on all leaf nodes. Each internal node except the root node connects to fewer leaf nodes and has less impact on the connected leaf nodes. Each value of the selected feature is multiplied by a number l_s, which is equal to the number of leaf nodes it connected to in the decision tree. It is represented as follows.

$$x_{is_{new}} = x_{is_{ori}} \times l_s, \quad s = 1, 2, \ldots, n \tag{6}$$

By this step, we enhance the impact of the selected features on leaf nodes based on the tree structure.

Mapping Layer. In the mapping layer, inspired by the entropy net [42], we map the decision tree to the ELM. Table 1 shows the corresponding mapping rules between nodes in decision tree and neurons in ELM.

Table 1: Mapping rules between DT and ELM.

DT	ELM
Internal nodes	Input neurons
Terminal nodes	Hidden neurons
Distinct classes	Output neurons
Paths between nodes	Connections between input layer and hidden layer

The number of internal nodes in decision tree equals the number of neurons in the input layer of ELM. Each leaf node in decision tree is mapped to a corresponding neuron in the hidden layer of ELM. The number of distinct classes in decision tree equals the number of neurons in the output layer of ELM. The paths between nodes in decision tree decide the connections between input layer and hidden layer of ELM. The result of this mapping principle determines the numbers

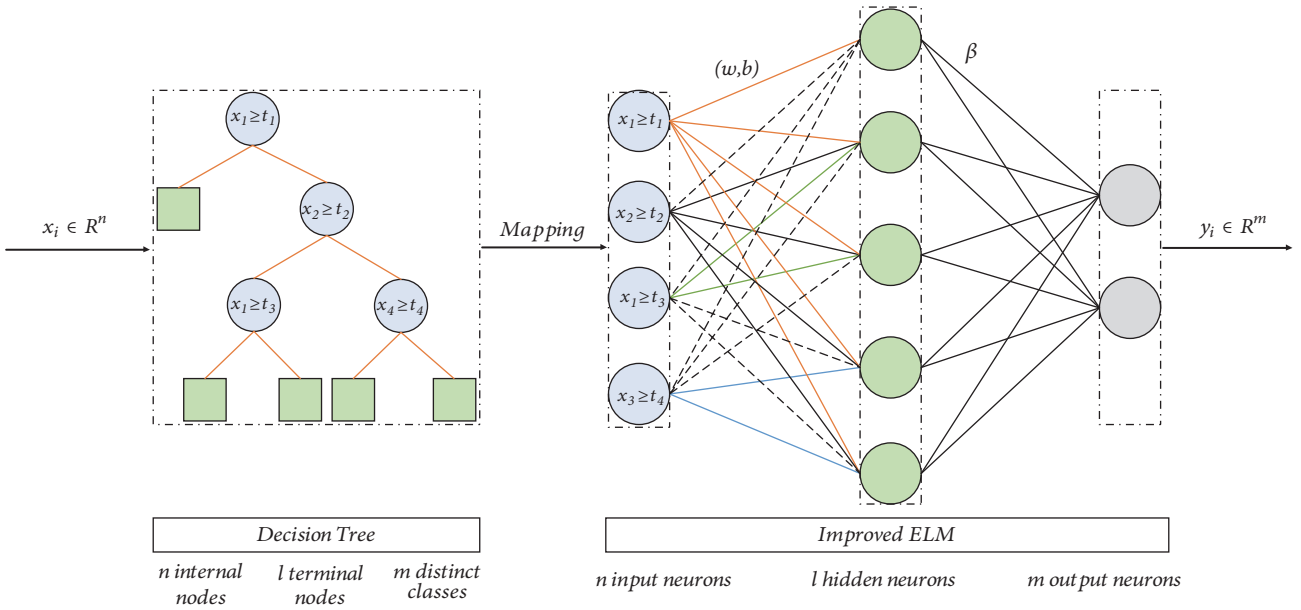

FIGURE 2: An illustration of mapping the decision tree to ELM. The internal nodes, leaf nodes, and distinct classes in the decision tree are mapped to the corresponding neurons in the input layer, hidden layer, and output layer of ELM, respectively. The paths between nodes determine the connections between input layer and hidden layer.

of neurons in each layer. Meanwhile, it improves ELM with fewer connections.

Figure 2 shows an illustration of mapping the decision tree to ELM. The first neuron of the input layer is mapped from the root node of the decision tree. It connects to all hidden neurons mapped from all leaf nodes. That means the first neuron has impact on every hidden neuron. The second neuron of the input layer connects to the four hidden neurons according to the decision tree structure. That means the second neuron has impact on the four hidden neurons. The dashed lines show that there exist no corresponding paths between the internal nodes and leaf nodes in the decision tree. Therefore, there exist no connections between the corresponding neurons in the input layer and the corresponding neurons in the hidden layer of ELM.

Improved ELM. Once the structure of ELM is determined, the enhanced features are input into the ELM. The connectionless weights between input layer and hidden layer are initialized with zero or extremely small values very close to zero. Other connection weights as well as biases of the hidden layer are initialized randomly. Unique optimal solution can be obtained once the numbers of hidden neurons and initialized parameters are determined.

There are N random instances (x_i, y_i), where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$, $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^T \in R^m$. A SLFN with L hidden neurons can be represented as follows:

$$\sum_{j=1}^{L} \beta_j g\left(w_j \cdot x_i + b_j\right) = o_i, \quad i = 1, 2, \ldots, N \tag{7}$$

where $g(x)$ is the activation function of hidden neuron. $w_j = [w_{j1}, w_{j2}, \ldots, w_{jn}]^T$ is the weight vector of input neurons connecting to i_{th} hidden neuron. The inner product of w_j and x_i is $w_j \cdot x_i$. The bias of the j_{th} hidden neuron is b_j. The weight vector of the j_{th} hidden neuron connecting to the output neurons is $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^T$.

The target of a standard SLFN is to approximate these N instances with zero error which is represented as (8), where the desired output is o_i and the actual output is y_i.

$$\sum_{i=1}^{N} \left\| o_i - y_i \right\| = 0 \tag{8}$$

In other words, there exist proper w_j, b_j, and β_j such that

$$\sum_{j=1}^{L} \beta_j g\left(w_j \cdot x_i + b_j\right) = y_i, \quad i = 1, 2, \ldots, N. \tag{9}$$

Equation (9) can be represented completely as follows.

$$H\beta = Y \tag{10}$$

where

$$H = \begin{bmatrix} g\left(w_1 \cdot x_1 + b_1\right) & \cdots & g\left(w_L \cdot x_1 + b_L\right) \\ \vdots & \vdots & \vdots \\ g\left(w_1 \cdot x_N + b_L\right) & \cdots & g\left(w_L \cdot x_N + b_L\right) \end{bmatrix}_{N \times L}, \tag{11}$$

1: Give the training set $T = (x_i, y_i) \mid x_i \in R^n,\ y_i \in R^m$, activation function $g(x)$, number of hidden neurons L.
2: Randomly assign input weight vector w_j and the bias b_j except the connectionless weights and biases between input layer and hidden layer with zero.
3: Calculate the hidden layer output matrix H.
4: Calculate the output weight vector $\beta = \mathrm{H}^{\dagger}Y$ where $\mathrm{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H.
5: Obtain the predicted values based on the input variables.

Algorithm 2: Improved ELM.

Table 2: Dataset description.

Information	Description
Enrollment	Records denoting what course each learner has enrolled (120,542 entries)
Object	Relationships of courses and modules
Log	Learning behavior records (8,157,277 entries)
Date	The start and end time of courses
Truth	Labels indicating whether learners dropout or complete the whole courses

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L\times M}, \tag{12}$$

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N\times M} \tag{13}$$

Once w_j and b_j are determined, the output matrix of the hidden layer H is uniquely calculated. The training process can be transformed into finding a least-squares solution of the linear system in (9). The algorithm can be described as in Algorithm 2.

3. Experimental Results

3.1. Dataset and Preprocess. The effectiveness of our proposed algorithm is tested on the benchmark dataset KDD 2015 which contains five kinds of information. The detailed description of the dataset is shown in Table 2. From the raw data, we define and extract several behavior features. The description is shown in Table 3.

The next step is to label each record according to the behavior features. If a learner has activities in the coming week, the dropout label is 0. Otherwise the dropout label is 1. A learner may begin learning in the later week but not the first week, and in the first several weeks, the learner will not be labeled as dropout; the week when the learner begins learning is seen as the first actually learning week for this learner. The first several weeks data will be deleted and then other weeks data will be labeled.

Table 3: Extracted behavior features.

Features	Description
x_1	The number of participating objects of course per week.
x_2-x_8	The behavior numbers of access, page close, problem, video, discussion, navigating, wiki per week.
x_9-x_{10}	The total and average numbers of all behaviors per week.
x_{11}	The number of active days per week.
x_{12}	The time consumption per week.
x_{13}-x_{16}	The behavior numbers of access, page close, problem, video from browser per week respectively.
x_{17}-x_{21}	The behavior numbers of access, discussion, problem, navigating, wiki from server per week respectively.
x_{22}-x_{23}	The numbers of all behaviors from browser and server per week respectively.

3.2. Experimental Setting and Evaluation. Experiments are carried out in MATLAB R2016b and Python 2.7 under a desktop computer with Intel 2.5GHz CPU and 8G RAM. The LIBSVM library [43] and Keras library [44] are used to implement the support vector and LSTM, respectively.

In order to evaluate the effectiveness of the proposed algorithm, accuracy, area under curve (AUC), F1-score, and training time are used as evaluation criteria. Accuracy is the proportion of correct prediction including dropout and retention. Precision is the proportion of dropout learners predicted correctly by the classifier in all predicted dropout learners. Recall is the proportion of dropout learners predicted correctly by the classifier in all real dropout learners. F1-score is the harmonic mean of precision and recall.

AUC depicts the degree to which a classifier makes a distinction between positive and negative samples. It is invariant to imbalanced data [45]. The receiver operating characteristics (ROC) plot the trained classifier's true positive rate against the false positive rate. The AUC is the integral over the interval $[0, 1]$ of the ROC curve. The closer the number to 1, the better the classification performance.

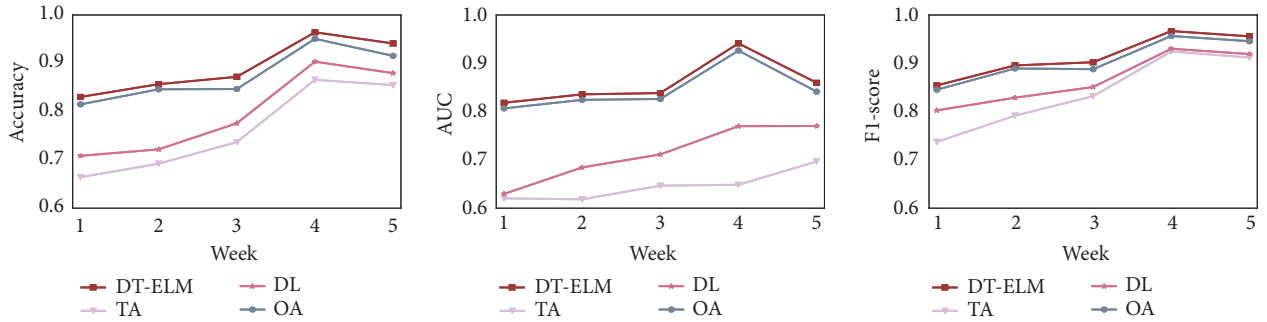

Figure 3: The overall performance of DT-ELM, traditional machine learning algorithms (TA), deep learning (DL), and optimization algorithm (OA) weekly. DT-ELM is about 12.78%, 22.19%, and 6.87% higher than baseline algorithms in terms of overall accuracy, AUC, and F1-score, respectively.

Table 4: The overall average training time (s) of DT-ELM, traditional machine learning algorithms (TA), deep learning algorithm (DL), and optimization algorithm (OA).

Algorithms	*DT-ELM*	TA	DL	OA
Time	*1.6383*	2.427	>100	3.3649

3.3. Overall Performance. We choose ten courses for experiments. The enrollment number ranges from several hundreds to about ten thousands. We divide the baselines into three categories, separately: traditional machine learning, deep learning, and optimization algorithm. Traditional machine learning algorithms include logistic regression, support vector machine, decision tree, back propagation neural network, and entropy net. LSTM is adopted as the deep learning algorithm. Genetic algorithm and ELM (GA-ELM) are combined as the optimization algorithm aiming to improve the ELM.

The results of overall performance in terms of accuracy, AUC, and F1-score are shown in Figure 3. The results of overall average training time are shown in Table 4.

Although there exists a wide range of course enrollments, the proposed DT-ELM algorithm performs much better than the three categories of baseline algorithms. DT-ELM is 89.28%, 85.86%, and 91.48% and about 12.78%, 22.19%, and 6.87% higher than baseline algorithms in terms of overall accuracy, AUC, and F1-score, respectively.

To be specific, the traditional machine learning algorithm performs the worst in terms of the three criteria. The results of the deep learning algorithm are much better. However, the deep learning algorithm has the longest training time. Although the optimization algorithm performs better than the deep learning algorithm, it does not perform as good as DT-ELM. DT-ELM performs the best in terms of accuracy, AUC, and F1-score. Meanwhile, it requires the least training time due to noniterative training process. The results have proved that DT-ELM reaches the goal of dropout prediction accurately and timely.

Another conclusion is that the last two weeks get better performance than the other weeks. To identify the reason, we make a statistical analysis of dropout rate weekly. We find that, compared to the first three weeks, the average dropout rate of the last two weeks of courses is higher. It means the behavior of learners is more likely to follow a pattern. On the other hand, it also illustrates the importance of dropout prediction. The dropout rates in the later stage of courses are higher than the initial stage generally. So it is better to find at-risk learners early in order to make effective interventions.

3.4. Impact of Feature Selection. To verify the effectiveness of feature selection, we make a comparison between DT-ELM and ELM. The results are shown in Figure 4. DT-ELM is about 2.78%, 2.87%, and 2.41% higher than ELM in terms of accuracy, AUC, and F1-score, respectively. It proves that feature selection has promoted the prediction results. Choosing as many features as possible may not be appropriate for dropout prediction. According to the entropy theory mentioned previously, features with different gain ratios have different classification ability. Features with low gain ratios may weaken the classification ability.

Although each course has different behavior features, two conclusions can be obtained. The average number of selected features is 12, which is less than the number of extracted features. It proves that using fewer features for dropout prediction can achieve better results than using all extracted features. Moreover, we find that discussion, active days, and time consumption are the three most important factors affecting prediction results.

3.5. Impact of Feature Enhancement and Connection Initialization. To verify the impact of feature enhancement, we make a comparison between DT-ELM and itself without feature enhancement (Without-FE). Similarly, to verity the impact of connection initialization, we make a comparison between DT-ELM with itself without connection initialization (Without-IC). The results of the three algorithms are shown in Figure 5.

The results of Without-FE and Without-IC are not as good as DT-ELM. DT-ELM is about 0.94%, 1.21%, and 0.9% higher than Without-IC in terms of accuracy, AUC, and F1-score, respectively. It is also about 2.13%, 2.25%, and 1.98% higher than Without-FE. The values of Without-IC are higher than Without-FE in terms of the three criteria, which indicates that feature enhancement plays a more important

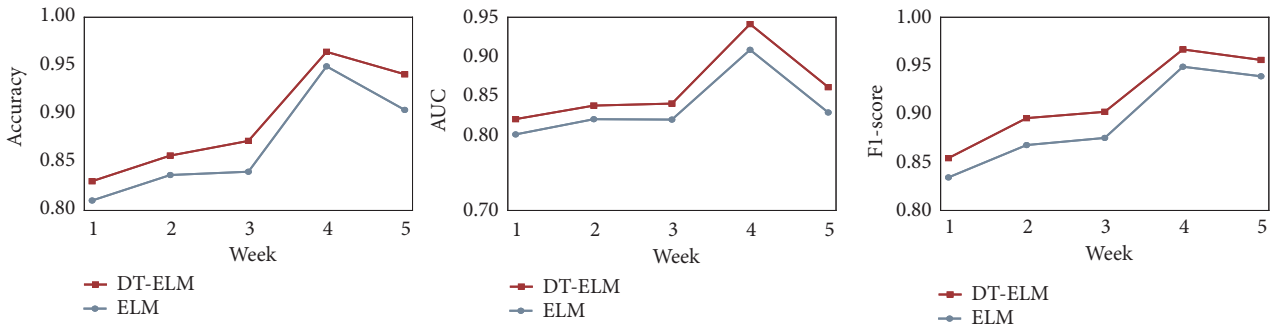

Figure 4: Impact of feature selection. DT-ELM is about 2.78%, 2.87%, and 2.41% higher than ELM in terms of accuracy, AUC, and F1-score, respectively.

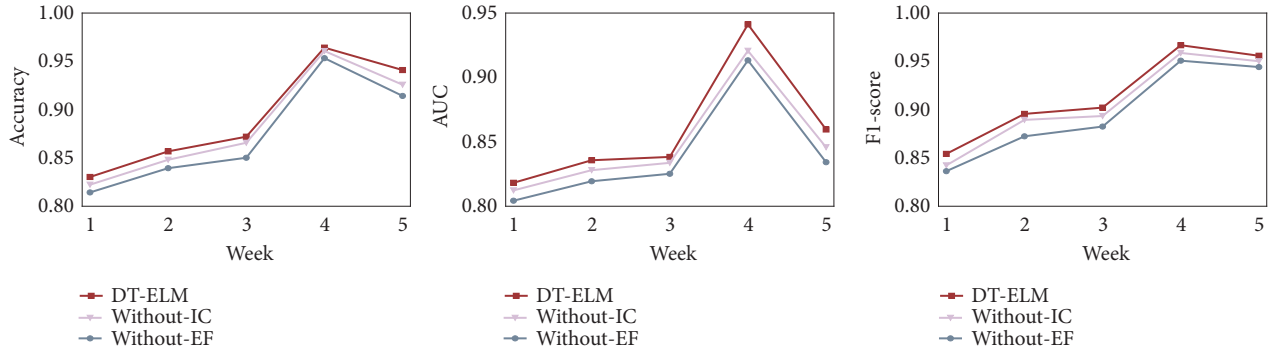

Figure 5: Impact of feature enhancement and connection initialization. Without-IC means DT-ELM without connection initialization and Without-FE means DT-ELM without feature enhancement. The results indicate that feature enhancement plays a more important role than connection initialization.

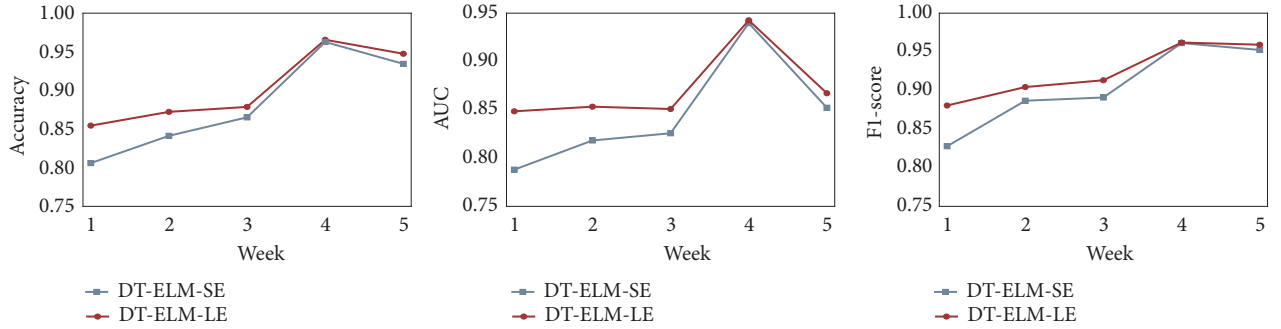

Figure 6: Comparison of different scales of enrollment in terms of accuracy, AUC, and F1-score. DT-ELM-SE contains courses with smaller numbers of enrollments and DT-ELM-LE contains courses with larger numbers of enrollments.

role than connection initialization. That is because features with better classification ability are enhanced depending on how much impact each feature has on other neurons.

3.6. Comparison of Different Numbers of Enrollments. To verify the effectiveness of DT-ELM on different numbers of enrollments, two groups of experiments are conducted and the results are shown in Figure 6. The first group (DT-ELM-SE) contains courses with smaller numbers of enrollments ranging from several hundreds to about one thousand. The second group (DT-ELM-LE) contains courses with larger numbers of enrollments ranging from several thousands to about then thousands.

Generally, the more the data used for training, the better the classification results. DT-ELM-LE is about 2.59%, 3.41%, and 2.54% higher than DT-ELM-SE in terms of accuracy, AUC, and F1-score, respectively. The average training time of DT-ELM-SE and DT-ELM-LE is 1.6014s and 1.6752s, respectively. Although courses with less enrollments achieve lower values than courses with more enrollments for the three criteria, the proposed algorithm still performs better than the other algorithms. That means dropout can be predicted

Table 5: The average accuracy of different algorithms weekly, including logistic regression (LR), support vector machine (SVM), back propagation neural network (BP), decision tree (DT), entropy net (EN), LSTM, ELM, and GA-ELM.

Week	Algorithms								
	DT-ELM	LR	SVM	BP	DT	EN	LSTM	ELM	GA-ELM
Week1	*0.8303*	0.6633	0.5643	0.6992	0.6876	0.7067	0.7067	0.8103	0.8149
Week2	*0.8568*	0.686	0.6581	0.705	0.6957	0.7157	0.7218	0.8367	0.8462
Week3	*0.872*	0.7284	0.7218	0.7443	0.7325	0.757	0.7758	0.8401	0.8467
Week4	*0.9642*	0.8726	0.827	0.8892	0.8627	0.8773	0.8773	0.9492	0.9507
Week5	*0.941*	0.8498	0.8618	0.8447	0.8526	0.8656	0.8656	0.9041	0.9154

Table 6: The average AUC of different algorithms weekly.

Week	Algorithms								
	DT-ELM	LR	SVM	BP	DT	EN	LSTM	ELM	GA-ELM
Week1	*0.8182*	0.632	0.6272	0.6189	0.5979	0.6256	0.6295	0.7984	0.8066
Week2	*0.8357*	0.6255	0.5914	0.6216	0.6243	0.6264	0.6844	0.8181	0.8243
Week3	*0.8383*	0.6527	0.6109	0.6586	0.6438	0.667	0.7113	0.8176	0.8261
Week4	*0.9412*	0.7103	0.6454	0.6117	0.6507	0.6246	0.7698	0.9079	0.9263
Week5	*0.8596*	0.6652	0.6918	0.7148	0.6954	0.7166	0.7701	0.8268	0.8413

Table 7: The average F1-score of different algorithms weekly.

Week	Algorithms								
	DT-ELM	LR	SVM	BP	DT	EN	LSTM	ELM	GA-ELM
Week1	*0.8542*	0.7378	0.5907	0.7719	0.798	0.788	0.8025	0.8342	0.8453
Week2	*0.8956*	0.7913	0.7341	0.8091	0.8	0.8238	0.8287	0.8677	0.8893
Week3	*0.9021*	0.8264	0.8107	0.8392	0.8321	0.8511	0.8508	0.8751	0.8878
Week4	*0.9667*	0.9315	0.8982	0.9348	0.9222	0.9358	0.9301	0.9488	0.9565
Week5	*0.9558*	0.9118	0.9185	0.9092	0.9149	0.905	0.9191	0.9389	0.9457

accurately and timely in courses with different numbers of enrollments.

3.7. Performance with Different Algorithms. The detailed results of different algorithms are shown in Tables 5–7. Observing the results, logistic regression and support vector machine achieve lower values in accuracy, AUC, and F1-score than the other algorithms. Back propagation neural network and decision tree perform better than logistic regression and support vector machine. Entropy net utilizes decision tree to optimize performance, and it performs better than back propagation neural network. Different from entropy net, we utilize decision tree to optimize ELM. Although the performance of LSTM is much better than the traditional algorithms, it is time intensive based on previous results.

It is obvious that ELM-based algorithms perform better than other algorithms. That is because ELM can get the smallest training error by calculating the least-squares solutions of the network output weights. GA-ELM achieves good results due to its function of feature selection. However, it also needs iterative training and lacks structure initialization of the ELM. DT-ELM optimizes the structure of ELM and performs much better than ELM and GA-ELM.

4. Discussion

Sufficient experiments are designed and implemented from various perspectives. For the overall performance, it achieves the best performance compared to different algorithms. Besides, we also explain why the last two weeks get better performance than the other weeks. The experimental results of feature selection demonstrate the importance of the feature selection. The reason is that features with different gain ratios have different classification ability, which helps get a higher performance. Feature enhancement and connection initialization both contribute to results, and feature enhancement plays a more important role than connection initialization due to its higher promotion on three criteria. The results of different numbers of enrollments prove the universality of our algorithm.

The experimental results have proved the effectiveness and universality of our algorithm. However, there is still an important question that needs to be considered. Do the instructors need to make interventions for all dropout learners? Our goal is to find the at-risk learners who may stop learning in the coming week and help instructors to take corresponding interventions. From the perspective of behavior, break means dropout. However, interventions

would not be taken for all dropout learners, because while making interventions, besides the behavior factor, some other factors [46], such as age, occupation, motivation, and learner type [47], should be taken into consideration. Our future work is to make interventions for at-risk learners based on learners' behaviors and background information.

5. Conclusion

Dropout prediction is an essential prerequisite to make interventions for at-risk learners. To address this issue, we propose a hybrid algorithm which combines the decision tree and ELM, which successfully settles the unsolved problems, including behavior discrepancy, iterative training, and structure initialization.

Compared to the evolutionary methods, DT-ELM selects features based on the entropy theory. Different from the neuron network based methods, it can analytically determine the output weights without updating parameters by repeated iterations. The benchmark dataset in multiple criteria demonstrates the effectiveness of DT-ELM and the results show that our algorithm can make dropout prediction accurately.

Data Availability

The dataset used to support this study is the open dataset KDD 2015 which is available from http://data-mining.philippe-fournier-viger.com/the-kddcup-2015-dataset-download-link/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research for this paper is supported by National Natural Science Foundation of China (Grant no. 61877050) and open project fund of Shaanxi Province Key Lab of Satellite and Terrestrial Network Tech, Shaanxi province financed projects for scientific and technological activities of overseas students (Grant no. 202160002).

References

[1] M. Vitiello, S. Walk, D. Helic, V. Chang, and C. Guetl, "Predicting dropouts on the successive offering of a MOOC," in *Proceedings of the 2017 International Conference MOOC-MAKER, MOOC-MAKER 2017*, pp. 11–20, Guatemala, November 2017.

[2] M. Vitiello, S. Walk, V. Chang, R. Hernandez, D. Helic, and C. Guetl, "MOOC dropouts: a multi-system classifier," in *European Conference on Technology Enhanced Learning*, pp. 300–314, Springer, 2017.

[3] E. J. Emanuel, "Online education: MOOCs taken by educated few," *Nature*, vol. 503, no. 7476, p. 342, 2013.

[4] G. Creeddikeogu and C. Carolyn, "Are you mooc-ing yet? a review for academic libraries," *Kansas Library Association College & University Libraries*, vol. 3, no. 1, pp. 9–13, 2013.

[5] S. Dhawal, "By the numbers: Moocs in 2013," Class Central, 2013.

[6] S. Dhawal, "By the numbers: Moocs in 2018," Class Central, 2018.

[7] H. Khalil and M. Ebner, "Moocs completion rates and possible methods to improve retention - a literature review," in *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2014.

[8] D. F. O. Onah, J. E. Sinclair, and R. Boyatt, "Dropout rates of massive open online courses: Behavioural patterns," in *International Conference on Education and New Learning Technologies*, pp. 5825–5834, 2014.

[9] R. Rivard, Measuring the mooc dropout rate, Inside Higher Ed, 8, 2013, 2013.

[10] J. Qiu, J. Tang, T. X. Liu et al., "Modeling and predicting learning behavior in MOOCs," in *ACM International Conference on Web Search and Data Mining*, pp. 93–102, 2016.

[11] Y. Zhu, L. Pei, and J. Shang, "Improving video engagement by gamification: a proposed Design of MOOC videos," in *International Conference on Blended Learning*, pp. 433–444, Springer, 2017.

[12] R. Bartoletti, "Learning through design: Mooc development as a method for exploring teaching methods," *Current Issues in Emerging eLearning*, vol. 3, no. 1, p. 2, 2016.

[13] S. L. Watson, J. Loizzo, W. R. Watson, C. Mueller, J. Lim, and P. A. Ertmer, "Instructional design, facilitation, and perceived learning outcomes: an exploratory case study of a human trafficking MOOC for attitudinal change," *Educational Technology Research and Development*, vol. 64, no. 6, pp. 1273–1300, 2016.

[14] S. Halawa, "Attrition and achievement gaps in online learning," in *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, pp. 57–66, March 2015.

[15] K. R. Koedinger, E. A. McLaughlin, J. Kim, J. Z. Jia, and N. L. Bier, "Learning is not a spectator sport: doing is better than watching for learning from a MOOC," in *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, pp. 111–120, Canada, March 2015.

[16] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H. Gehlbach, "Forecasting student achievement in MOOCs with natural language processing," in *Proceedings of the 6th International Conference on Learning Analytics and Knowledge, LAK 2016*, pp. 383–387, UK, April 2016.

[17] C. Taylor, K. Veeramachaneni, and U. M. O'Reilly, "Likely to stop? predicting stopout in massive open online courses," Computer Science, 2014.

[18] C. Ye and G. Biswas, "Early prediction of student dropout and performance in MOOCs using higher granularity temporal information," *Journal of Learning Analytics*, vol. 1, no. 3, pp. 169–172, 2014.

[19] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, "Predicting MOOC dropout over weeks using machine learning methods," in *EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in Moocs*, pp. 60–65, October 2014.

[20] S. Nagrecha, J. Z. Dillon, and N. V. Chawla, "Mooc dropout prediction: Lessons learned from making pipelines interpretable," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 351–359, International World Wide Web Conferences Steering Committee, 2017.

[21] D. Peng and G. Aggarwal, "Modeling mooc dropouts," *Entropy*, vol. 10, no. 114, pp. 1–5, 2015.

[22] J. Liang, J. Yang, Y. Wu, C. Li, and L. Zheng, "Big data application in education: dropout prediction in edx MOOCs," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, pp. 440–443, IEEE, April 2016.

[23] G. Balakrishnan and D. Coetzee, "Predicting student retention in massive open online courses using hidden markov models," Electrical Engineering and Computer Sciences University of California at Berkeley, 2013.

[24] C. Lang, G. Siemens, A. Wise, and D. Gasevic, *Handbook of Learning Analytics*, Society for Learning Analytics Research (SoLAR), 2017.

[25] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in *IEEE International Conference on Data Mining Workshop*, pp. 256–263, USA, November 2015.

[26] W. Li, M. Gao, H. Li, Q. Xiong, J. Wen, and Z. Wu, "Dropout prediction in MOOCs using behavior features and multi-view semi-supervised learning," in *Proceedings of the 2016 International Joint Conference on Neural Networks, IJCNN 2016*, pp. 3130–3137, Canada, July 2016.

[27] M. Wen and C. P. Rosé, "Identifying latent study habits by mining learner behavior patterns in massive open online courses," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014*, pp. 1983–1986, China, November 2014.

[28] S. Lei, A. I. Cristea, M. S. Awan, C. Stewart, and M. Hendrix, "Towards understanding learning behavior patterns in social adaptive personalized e-learning," in *Proceedings of the Nineteenth Americas Conference on Information Systems*, pp. 15–17, 2003.

[29] J. H. Yang and V. Honavar, "Feature subset selection using genetic algorithm," *IEEE Intelligent Systems & Their Applications*, vol. 13, no. 2, pp. 44–48, 1998.

[30] A. K. Das, S. Das, and A. Ghosh, "Ensemble feature selection using bi-objective genetic algorithm," *Knowledge-Based Systems*, vol. 123, pp. 116–127, 2017.

[31] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, IEEE, July 2004.

[32] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[33] A. Ramesh, D. Goldwasser, B. Huang, H. Daumë, and L. Getoor, "Learning latent engagement patterns of students in online courses," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[34] J. He, J. Bailey, B. I. P. Rubinstein, and R. Zhang, "Identifying at-risk students in massive open online courses," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2005.

[35] D. Yang, T. Sinha, D. Adamson, and C. P. Rosé, "Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses," in *Proceedings of the 2013 NIPS Data-driven education workshop*, vol. 11, p. 14, 2013.

[36] M. Sharkey and R. Sanders, "A process for predicting MOOC attrition," in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pp. 50–54, Doha, Qatar, October 2014.

[37] F. Wang and L. Chen, "A nonlinear state space model for identifying at-risk students in open online courses," in *Proceedings of the 9th International Conference on Educational Data Mining*, vol. 16, pp. 527–532, 2016.

[38] A. Bhattacherjee, "Understanding information systems continuance: an expectation-confirmation model," *MIS Quarterly*, vol. 25, no. 3, pp. 351–370, 2001.

[39] Z. Junjie, "Exploring the factors affecting learners continuance intention of moocs for online collaborative learning: an extended ecm perspective," *Australasian Journal of Educational Technology*, vol. 33, no. 5, pp. 123–135, 2017.

[40] J. K. T. Tang, H. Xie, and T.-L. Wong, "A big data framework for early identification of dropout students in MOOC," *Communications in Computer and Information Science*, vol. 559, pp. 127–132, 2015.

[41] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[42] I. K. Sethi, "Entropy nets: from decision trees to neural networks," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1605–1613, 1990.

[43] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[44] C. François et al., Keras: The python deep learning library. Astrophysics Source Code Library, 2018.

[45] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley, "MOOC dropout prediction: How to measure accuracy?" in *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pp. 161–164, ACM, 2017.

[46] J. DeBoer, G. S. Stump, D. Seaton, and L. Breslow, "Diversity in mooc students backgrounds and behaviors in relationship to performance in 6.002 x," in *Proceedings of the Sixth Learning International Networks Consortium Conference*, vol. 4, pp. 16–19, 2013.

[47] P. G. de Barba, G. E. Kennedy, and M. D. Ainley, "The role of students' motivation and participation in predicting performance in a MOOC," *Journal of Computer Assisted Learning*, vol. 32, no. 3, pp. 218–231, 2016.