When and who at risk? Call back at these critical points

Yuntao Li
Department of Machine Intelligence
Key Laboratory of Machine Perception(MOE)
Peking University
li-yuntao@qq.com

Chengzhen Fu
Department of Machine Intelligence
Key Laboratory of Machine Perception(MOE)
Peking University
wallaceapril@163.com

Yan Zhang
Department of Machine Intelligence
Key Laboratory of Machine Perception(MOE)
Peking University
zhy@cis.pku.edu.cn

ABSTRACT
Since MOOC is suffering high dropout rate, researchers try to explore the reasons and mitigate it. Focusing on this task, we employ a composite model to infer behaviors of learners in the coming weeks based on his/her history log of learning activities, including interaction with video lectures, participation in discussion forum, and performance of assignments, etc.

The prediction accuracy of our proposed model outperforms related methods. Besides, we try combining the model with suggested interventions, such as sending reminder emails to at-risk learners. Future work, which is currently underway, will evaluate its influence on mitigating dropout rate.

Keywords
MOOC; dropout; Stacked Sparse Autoencoder; RNN

1. INTRODUCTION
Recently, online education, for which landmark concept is MOOCs (Massive Open Online Courses), has become a new global craze, bringing several MOOC platforms including EdX, Coursera, and Udacity, etc. Due to the freedom of time and place learning at MOOCs, a large scale of learners has been benefit from this new form of online learning. A typical course of MOOC lasts for 6-12 weeks, with learners of diverse backgrounds and major field. Besides, MOOC learners may have different intentions and motivations, causing their extents and leave for various reasons.

Despite the increasing popularity of MOOCs, the extremely low rate of completion has been considered from the beginning. Drop-out is concerned as one of the most critical problem of MOOCs. Drop-out indicates situations that a student registers a course, watches course materials, or even attends the quizzes, but eventually quits without attending the final test. It has been researched that an average completion rate of MOOCs comes as low as 7 percent, ranging from 0.8 percent in Princeton's (History of the World since 1300), to 19.2 percent in the "Functional Programming Principles in Scala" course [7]. MOOC platforms are facing a concerning issue due to a high learners' dropout rate.

Thus, identifying at-risk learners by predicting their dropout probability thus becomes timely important, given that early prediction can help instructors provide proper support to those learners to retain their learning interests aiming at guaranteeing them a regular process of study without doing a crash job or even dropout. Addressing this task, we focus on predicting learners' state for the next consecutive two weeks. We particularly formulate this issue as a multi-classification problem, and develop a Stacked Sparse Autoencoder (SSAE)+Softmax model to solve it. Essentially, our model has several advantages. First, it incorporates multiple features based on characterizing learners' weekly engagements on the MOOC platform. Second, it discovers correlations between observed explanatory features. The new compressed feature representation transformed by SSAE performs better than the previous one, based on the input of classifiers. Third, the model considers both the current and previous states to estimate the next states, which makes it more flexible to model students' dynamics.

By training a model to identify at-risk students, we can apply this model on online MOOC platforms, enabling it to calculate students' at-risk-rate regularly and send emails to them automatically. Hopefully some of these at-risk students will continue their learning.

We make contributions in this paper as follows:

1. We employ different composite models that incorporate multiple features to infer behavior in the coming weeks based on weekly history of learning data. The model is an end-to-end neural network model, which means it can be trained as a whole. Our results indicate that model of SSAE+Softmax performs best and achieve higher AUC score consistently, which is superior to the baseline SVM model.

2. We try combining the model with suggested interventions such as sending reminder emails to at-risk learners. Though we do not conduct real experiments of sending emails, the paper proposes a preliminary framework of applying experimental results to determining to whom reminder emails should be sent and when to send.

3. We explore to what extent each single feature can influence dropout probability and try to cluster dropout learners by employing k-means clustering algorithm, proving that features extracted from course engagements are effective indicators of which class a low-performing learner belongs to separated by their pattern of behaviors. Future work will shade light into the relationships between behavior patterns of learners and reasons why they quit the course.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the description of the dataset and features derived from the dataset. Section 4 introduces our model in detail. Experimental results and discussion are presented in Section 5 and 6. Finally, Section 7 concludes our work in this paper.

2. RELATED WORKS

Mitigating MOOC dropout rate is essential for boosting the values of MOOCs, thus the mechanisms that can predict student dropout become increasingly important.

Some exploratory analysis suggests that student behavior in the discussion forum helps predict attrition. Yang et al. [6] present a foundation for research investigating the social factors that affect dropout along the way during participation in MOOCs. To operationalize these factors, they define metrics related to posting behavior (thread starter, post length, content length) and social positioning (posts & replies) within the resulting reply network. Similarly, some researchers (Ramesh et al. [8]) explore other aspects of discussion forum such as viewing posts, sentiment. This perspective provides a potentially valuable source of insight for design of MOOCs that may be more conducive to social engagement that promotes commitment and therefore lower attrition. It is restrictive in application because it mainly lowers attrition of learners who drops out mainly because of hard interpersonal connection foundation online.

Many researchers aim at modeling learning behaviors over duration of weeks. Their pursuit is to extract significant features by parsing the clickstream file where each line represents a web request. These effective features include lecture interaction features, forum interaction features, assignment features [1–4,11], which capture the activity level of learners.

In terms of applied models, Kloft et al. [5] explore support vector machines (SVM) to predict the state of learners in the later phases of a course. Balakrishnan et al. [2] quantize the feature space into a discrete number of observable states that are integral to a Discrete Single Stream HMM. Fei et al. [9] propose recurrent neural network (RNN) model with long short-term memory (LSTM) cells.

3. DATA SET AND FEATURE SET

3.1 Dataset

The learner activity log data came from a publicly held data mining competition called KDD CUP 2015. It includes 79186 learners, each of whom enrolled in at least one course of the whole set of 39 courses. In total, the clickstream data includes 8,157,277 log records and the longest lifetime of enrollment is 6 weeks. Most of the data is user activity log data and course structure data.

3.2 Feature set

As stated above, our goal is to estimate the probability that a student stops engaging with a course for the next two weeks, given her/his learning activities up to the current time step.

The dropout probabilities are closely related to learners' engagements to courses, which are mainly characterized by design of forum, lecture and assessment features. To express the time-varying behaviors of learners, we extract 17 typical features of each week t for each learner i, denoted as vector $x_i^{(i)} \in R^17$, as presented in Table 1. It can be noticed that, features we selected are vital but highly correlated with each other, and we will introduce a model to cancel this redundancy.

Feature	Description
f1-f3	Number of posts in discussions, videos watched, problems attempted in week t respectively
f4-f6	Total number of discussions made, videos watched, problems attempted by week t
f7-f9	Average number of discussions, videos, problems attempted per week by week t
f10-f12	Average number of discussions, videos, problems attempted per session in week t
f13	Sum of number of another activities (navigate, access, page close, wiki) in week t
f14	Total number of activities in week t
f15	Total number of active days in week t
f16	Total number of time consumption in week t
f17	Total numbers of sessions in week t

Table 1: List of features derived for week t

3.2.1 Interactions with forums

A MOOC forum provides a platform to facilitate the communication between learners and lecturers. The more actively the learners interact with their partners, the more a learner feels she/he belongs in the course learning and the more likely she/he is to complete the learning tasks. Some features, such as viewing a post, receiving a reply, following a thread and up-voting, are strong indicators of engagement and sense of community [6, 7].

3.2.2 Interactions with lectures

Because the lecture videos are the most important learning resource for the learning participants, the video playing should be investigated, as done by other researchers. Among these works, Kim et al. [1] explored some click actions when watching videos. These behaviors can be classified into six types: skipping, zooming, playing, replaying, pausing, and quitting.

3.2.3 Interactions with assignments

It is reasonable to hypothesize that an active and engaged student would monitor their assignment a few times every week because material is released and due on a weekly basis. When monitoring this week by week, we can roughly estimate how far up-to-date a student is with a course. It is acknowledged that if a learner falls behind too much, it is hard to catch up and thus determination to complete is lost [2].

Furthermore, we observe from the user activity log data whether the learners are active in session, as the data contain multiple records in quick succession. We define the elapsed time of two separate sessions as 45 minutes. If the gap between a learner's two consecutive operation is more than 45 minutes, we assume that the learner quit and logged in again.

Consequently, for current week t, we obtain a sequence of $(x_1^{(i)}, x_2^{(i)}, ..., x_t^{(i)})$ for each learner i across t weeks and the corresponding sequence of dropout labels $(y_1^{(i)}, y_2^{((i))}, ..., y_{t-1}^{(i)})$. If there are activities associated with student i in the coming week, the dropout label in week t is assigned as $y_t(i) = 0$, otherwise, $y_t(i) = 1$. Notably, all features should be centered and normalized to unit standard deviation (mean of 0 and variance of 1).

4. OUR MODEL

4.1 Feature Extractor: Stacked Sparse Autoencoder (SSAE)

Now suppose that we have extracted weekly features from user activity log record, we employ a model named Stacked Sparse Autoencoder (SSAE) to discover high level representation of input features and correlations among them. In this part, we aim to produce a better feature representation that can show patterns of behavior for learners.

Autoencoder neural networks are a serial of models which can re-represent features by encoding them into a high level representation using a set of parameters and decode it back to its original values using another set of parameters. A sparse autoencoder neural network consists of an input layer, a hidden layer and an output layer, whose size of hidden layer is greater than its input layer. The network structure is presented in Figure 1.

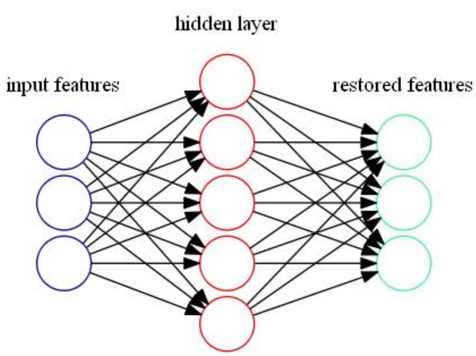

Figure 1: Network Structure

Formally, let the vector of input layer be the features of learner i extracted from weekly history of learning behavior features. We train the network to minimize the divergence between the input layer and the output layer, i.e., $h_{W,b}(x_t^{(i)}) = x_t^{(i)}$. After the model goes into convergence, which means it achieves a minimal difference between input features and output values, the hidden layer learns a new representation of the input. The numbers and dimensions of the hidden layer controls the complexity of the network and requires parameter values tuning to determine its optimal value. Notably, the new features that the hidden layer represents will be as the input of a classifier. To train this autoencoder network, we apply back-propagation algorithm to minimize overall cost function as follows:

$$J_{\text{sparse}}(W, b) = J(W, b) + \beta \sum W \cdot W$$

Where J (W, b) is calculated by two parts: an average sum-of-squares error and penalty term that helps prevent over fitting. $\sum W \cdot W$ means a sum of every element in matrix which is the element wised multiple of W. β represents weight of the sparsity penalty term.

Here we do not introduce the details; computational details can be found in [10].

In order to generate more general (higher-level-presented) features, we use a method called stacked to enrich capacity of our model. We train an autoencoder first and use its features as the input and output of another autoencoder. Thus we get a more abstract representation of original features which can be more suitable for describing learners' inner condition.

Compared with other methods like PCA, the neural network based SSAE is more strong. For most cases, relations between meta features are complex and can not be represented by simple functions like linear functions, thus traditional methods are not able to separate them well. However, neural networks have the ability to fit any function as long as it is given enough capacity(e.g. enough depth of layers of amount of cells), which ensures it to project meta features in an independent orthogonal linear space.

4.2 Sequenced feature combiner: RNN

A RNN (or Recurrent Neural Network) is a class of artificial neural networks dealing with sequence data. It takes sequenced data step by step, and generates an output according to all previous inputs on every step. A basic RNN with one hidden layer is shown in Figure 2.

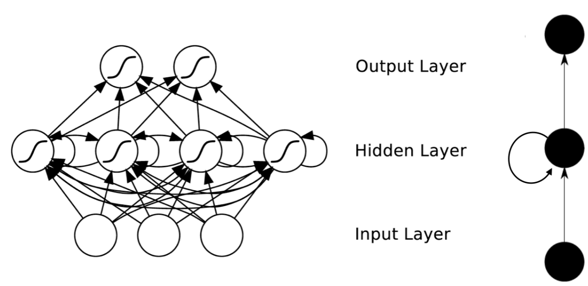

Figure 2: Basic RNN Structure

Formally, RNN is a function , where h is the hidden status (memory) of hidden units, and D is the size of input vector and L is the size of the output vector. The memory h changes every time while giving new inputs at each step.

The input vector of RNN is the high level representation generated by SSAE introduced in part 2. We aim to get a good feature representation, which can contain all learners' event histories within a fixed-length vector, to make prediction and classify dropout learners by his/her reason.

For a simple RNN, it has parameters $(W_h, U_h, W_y, b_h, b_y)$, where W_h controls what to absorb to memory from input features, and U_t determines what to remember and what to forget from the last memory status, and W_y sets the output value, and b_h and b_y are biases who make a global offset to both hidden status and output value.

The computational formula of this kind of RNN is shown below:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$
$$y_t = \sigma_y(W_y h_t + b_y)$$

where x_t and y_t represents input features and output vector at time t, and h_t is the memory hold by RNN. Here, σ_h and σ_y can be the same or different activation functions. Typical choices of activation functions are the sigmoid function and tanh function. Particularly, we choose tanh as activation function for both of the formulas. We will apply tanh in this paper as it typically yields to faster training (and sometimes also to better local minima). The operation tanh is calculated as follows:

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

We do not apply an LSTM used by other researchers [8] because of some reasons. An LSTM is a special kind of RNN who has the ability of forgetting, which means it can determine what to remember and when to forget its memory while getting new inputs, however, a simple RNN can only remember all its inputs. We think that, for a sequence no longer than six, forgetting should not be accepted. Besides, simple RNN requires less calculated quantities which makes it more suitable for a large scale online service.

4.3 Classifier

4.3.1 *Support Vector Machine(SVM)*

Some prior work mentioned in the related work inspires us to employ SVM to predict the learning state in the next consecutive two weeks. The SVM computes an affine-linear prediction function based on maximizing the margin of positive and negative examples:

$$\begin{aligned}(w, b) :=& argmin_{w,b}\frac{1}{2}||w||^2 \\ &+ C\Sigma_{i=1}^{n} max(0, 1 - y_i(< w, x > +b))\end{aligned}$$

After extracting features, we try to predict by using SVM and compare with results from Softmax. As there is distinct difference between dropout users and non-dropout users, therefore, we use the method of random sampling to confine the amount of these users into a comparatively small one. With this done, the model we gain will not cause overfitting to either classification.

With learning feature of current week obtained in 'Feature set' Section as input, we apply SVM to predict whether to drop out at the end of this week. Three Kernel Functions: linear, rbf and mlp are tried, and the prediction accuracy is estimated via 5-folds cross validation.

4.3.2 *Softmax Regression*

In the softmax regression setting, we are interested in multi-class classification (as opposed to only binary classification). It is expected to classify learners into three cases, which can be represented as $\{(0,0),(0,1),(1,1)\}$, where 1 means dropout, and the first number depends on whether to drop out after one week, the latter indicates results after two weeks. In this case, the label set can take on 3 different values, letting the predicted outcome for i-th learner $\in \{1, 2, 3\}$.

We aim to estimate the probability of the class label taking on each of the 3 different possible values of each learner. Thus, our hypothesis will output a 3-dimensional vector (whose elements sum to 1) giving us our estimated 3 probabilities. Concretely, our hypothesis takes the form:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)};\theta) \\ p(y^{(i)} = 2|x^{(i)};\theta) \\ p(y^{(i)} = 3|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{3} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ e^{\theta_3^T x^{(i)}} \end{bmatrix}$$

Where $\theta_1, \theta_2, \theta_3 \in R^n$ represent model parameters of softmax, and $\sum_{j=1}^{3} e^{\theta_j^T x^{(i)}}$ generalizes the probability distribution, leading to the sum of all the probability is 1.

5. EXPERIMENTS

5.1 AUC Score

We can observe from the KDD cup's label set that the labels are displayed with 79% positives and 21% negatives. Due to class imbalance phenomenon, accuracy is not a good metric. Instead, Area under receiver operating characteristic curve (ROC AUC) is the main metric we use to do parameter tuning and model selection. Furthermore, AUC measures how likely a classifier can correctly discriminate between positive and negative samples.

	Week 1	Week 2	Week 3	Week 4	Week5
SSAE+ Softmax	0.924	0.895	0.887	0.803	0.754
SSAE+ SVM	0.894	0.867	0.849	0.784	0.729
SVM	0.831	0.826	0.817	0.749	0.698

Table 2: AUC comparison of SSAE+Softmax, SSAE+SVM, SVM

Table 2 presents the average AUC scores across weeks by applying two different classifiers (Softmax, SVM). The results indicate that the models that employ SSAE to discover correlations among initial features extracted from dataset, such as SSAE + Softmax, SSAE + SVM, are more competitive. They are superior to the baseline SVM model and achieve higher AUC score consistently. For instance, for the first week, the AUC score of SSAE+SVM is 0.894, which is 7.58% improvement relative to that of SVM.

Specifically, we can observe that our proposed model SSAE + Softmax outperforms the other models across different weeks. The observation implies that Softmax performs consistently better than SVM in terms of classifying a learner's previous states and predicting whether he will drop out.

More notably, the AUC score decreases with increasing lifetime of the course. We infer that there might be more uncertainties related with dropout behavior that our model could not discover only from weekly history records. External forces such as lack of free time may result in more complex patterns of behavior. For instance, a learner may leave suddenly at week 4, while all statistical features of the previous three weeks strongly indicate he is not inclined to drop out.

5.2 Confusion Matrix

In this two class classification problem, the confusion matrix is a matrix with 4 entries, true positive(TP), false negative(FN), false positive(FP), and true negative(TN).

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = TruePositiveRate = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The comparisons of metric mentioned above are presented in Table 3. Model of SSAE+Softmax outperforms the other models consistently, proving good implement of the prediction task. It is convincing that the results across weeks lay a foundation to identify patterns of behavior and suggest interventions for inactive learners.

Model	Precision	Recall	F1 score
SSAE+Softmax	0.891	0.942	0.916
SSAE+SVM	0.873	0.907	0.890
SVM	0.854	0.887	0.870

Table 3: Performance comparison of SSAE+Softmax, SSAE+SVM, SVM

6. DISCUSSION

Experimental results of a real-world dataset demonstrate that dropout probability is consistently predicable across weeks for different students. The next step in applying the newly proposed model (SSAE+Softmax) to MOOC platforms aims to mitigate dropout rate by suggesting interventions, such as sending reminder emails, with the goal of informing at-risk learners to retain interests.

Email is a very cheap medium to reach learners and create awareness quickly. Our proposed model will contribute to determining to whom an email should be sent and when to send. Identifying at-risk learners precisely avoids bombarding active learners with unnecessary emails and at the same time informs them in time to call back as many of them as possible.

Here we only present a preliminary framework for sending reminder emails. Specifically, at the end of week t, first, we extract weekly feature vectors for t weeks and employ SSAE+Softmax to predict future states y_t and y_{t+1}. Then, we determine a candidate set of potential at-risk learners who satisfy y_t=1 and $y_{t+1} = 1$ where y_t means status of the next week. Finally, we observe her/his behavior in the coming week $t + 1$ for every selected learner. If the 'at risk' state is confirmed ($y_t = 1$), the platform will send reminder emails at the end of week $t + 1$ immediately.

Although the experiments presented in this paper are limited to KDD Cup, we plan to augment our model and evaluate the effectiveness of sending reminder emails in a real MOOC platform established by our university. Future work applying this model is currently underway and the idea for sending emails will be improved step by step.

With features observed as stated in Section 3, we finish the analysis of predicting dropout based on model mentioned in Section 4. After gauging the goodness of model performance, it is persuadable that we have the ability of predicting and diagnosing dropout. In the following part, we analyze how each feature could influence final dropout probability by conducting sensitivity analysis, and try to cluster dropout learners to figure out their patterns of behavior by applying k-means algorithm.

In order to make data comparable, we separate user events by different courses and take the course with the most students (which is also the one with the most accomplished students) as our studying example. First, we try to find out standard learner behaviors of those who accomplish the course with a good quality. We simply take all non-dropout students' event logs and take an average on each of the features, and regard this as a medial requirement for finish this course. Next, we change each of the features step by step and make prediction using our neural networks with fixed parameters, and then we get three outputs representing probabilities of dropout in one or two weeks, or not dropout. We evaluate a score ranging from 0 to 1 to evaluate quality of these features.

Algorithm 1 Univariate analysis of feature_i

```
procedure UnivariateAnalysis(model, input_features)
    standard ← average(input_feature)
    for rate ∈ (0.5 ... 1.5) do
        features ← standard
        features_i ← rate × feature_i;
        EvaluateDropoutRate(model, features);
    end for
end procedure
```

In Algorithm 1, "input-features" are features of those complete the courses, and "model" is the model we introduced above using SSAE, RNN and Softmax to predict a dropout rate, which is regraded as a score ranging from 0 to 1.

Notably, these features representing learning behaviors are classified into two categories: those related to course materials directly (e.g., watching videos, browsing wiki) and those not (e.g., navigate, page_close). We test some features to show how they influence a learner's dropout probability, as presented in Figure 3.

When times of watching video is 60 percent the amount of the standard statistic, the dropout probability increases sharply from 0.12 to 0.875. In this case, the dropout probability for feature page_close increases from 0.52 to 0.774, less significantly. It implies that, metrics closely related to course materials matter more than the others. Compared to indirect activities, times of direct engagements with course

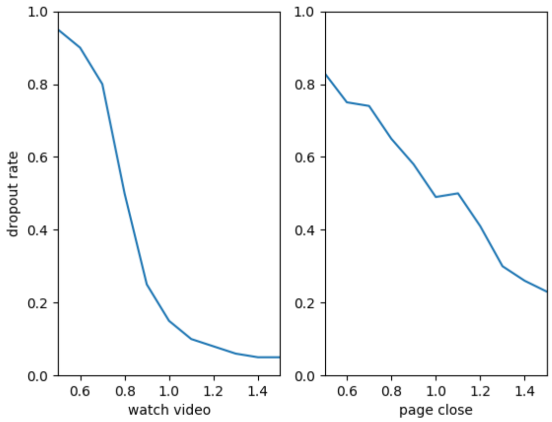

Figure 3: Sensitivity Analyses

materials are highly relevant to probability of accomplishing the course.

We then try to cluster dropout learners by employing k-means clustering algorithm, in which we set $k = 10$. Features extracted in Section 3 are effective indicators of which pattern of behavior a low-performing learner belongs to. We map any feature vector to one of the 10 clusters. There are two clusters whose number of low-performing learners are apparently larger than the others.

Inactive learners belonging to one cluster mentioned above preform worse with increasing lifetime of engagements. By monitoring their learning behavior in terms of lecture video, discussion and assignments, we find the numbers decrease week by week significantly. It can be inferred that they are putting less and less effort into learning as the course continues, which is a great indicator of failing to keep up with the pace of the course.

Inactive learners belonging to another cluster display a complex pattern of behavior. For instance, they leave the course for one or two weeks and then come back to learn. At the beginning, these learners display a high level of perseverance and self-discipline. Almost all the statistics demonstrate that they have regular patterns of studying, which can be confirmed by low dropout probability computed by our model. However, they behaved poorly in the coming weeks. Specifically, for some learners, the number of video watched, discussion made, and problems attempted all reach 0 suddenly. After some weeks, these learners come back to learn. Meanwhile, all learning data reaches the highest in comparison with previous weeks. Finally, they don't take exams and drop out. It may be inferred that such learners are "trying but not succeeding", due to the limit of time allowance (maybe other external forces).

In the future, to extend our model, we will send those learners predicted to leave the course a survey to find out why they are disengaging. We will shade light into the relationships between behavior patterns of learners and reasons why they quit the course.

7. CONCLUSIONS

In this paper, we propose different composite models that incorporate multiple features to infer behavior for the next two weeks based on features extracted from weekly history of learning data. The SSAE+Softmax model achieves a higher AUC score consistently, being superior to the baseline SVM model. Besides, application of the model including an automated email reminder system is under construction.

8. ACKNOWLEDGEMENT

This work is supported by NSFC under Grant No. 61532001 and 61370054, and MOE-RCOE under Grant No. 2016ZD201.

9. REFERENCES

[1] Kim, J., Guo, P. J., Seaton, D. T., Mitros, P., Gajos, K. Z., & Miller, R. C. (2014, March). Understanding in-video dropouts and interaction peaks inonline lecture videos. In Proceedings of the first ACM conference on Learning@ scale conference (pp. 31-40). ACM.

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

[3] Halawa S, Greene D, & Mitchell J. Dropout prediction in MOOCs using learner activity features[J]. Experiences and best practices in and around MOOCs, 2014, 7.

[4] He, J., Bailey, J., Rubinstein, B. I., & Zhang, R. (2015, January). Identifying At-Risk Students in Massive Open Online Courses. In AAAI (pp. 1749-1755).

[5] Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014, October). Predicting MOOC dropout over weeks using machine learning methods. In Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs (pp. 60-65).

[6] Yang, D., Sinha, T., Adamson, D., & RosÃľ, C. P. (2013, December). 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).

[7] Rachelle Peterson. 2013. Why Do Students Drop Out of MOOCs? Article. (13 November 2013.). https://www.nas.org/articles/why_do_students_drop_out_of_moocs

[8] Ramesh, A., Goldwasser, D., Huang, B., DaumÃľ III, H., & Getoor, L. (2014, July). Learning latent engagement patterns of students in online courses. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (pp. 1272-1278). AAAI Press.

[9] Fei, M., Yeung, & D. Y. (2015, November). Temporal Models for Predicting Student Dropout in Massive Open Online Courses. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (pp. 256-263). IEEE.

[10] Ng, A. (2011). Sparse autoencoder. CS294A Lecture notes, 72, 1-19.

[11] Henrie, C. R., Halverson, L. R., & Graham, C. R. (2015). Measuring student engagement in technology-mediated learning. Elsevier Science Ltd.