MOOC Dropout Prediction Based on Dynamic Embedding Representation Learning

Lin Wang
Science and Technology on Information Systems Engineering Laboratory, National University of Defense and Technology College of Systems Engineering,, Changsha, China,
jxwanglin@tju.edu.cn

Zhengfei Yu
Science and Technology on Information Systems Engineering Laboratory, National University of Defense and Technology College of Systems Engineering,, Changsha, China,
yuzhengfei19@nudt.edu.cn

Mengru Wang
Science and Technology on Information Systems Engineering Laboratory, National University of Defense and Technology College of Systems Engineering,, Changsha, China,
mengruwg@163.com

Xixi Zhu
Science and Technology on Information Systems Engineering Laboratory, National University of Defense and Technology College of Systems Engineering,, Changsha, China,
zhuxixi14@nudt.edu.cn

Yun Zhou*
Science and Technology on Information Systems Engineering Laboratory, National University of Defense and Technology College of Systems Engineering,, Changsha, China,
zhouyun@nudt.edu.cn

ABSTRACT

Massive Open Online Courses (MOOCs) received great attentions in recent years. Most MOOCs have huge number of participants, which usually introduce another challenge—the extremely high dropout rate. Thus, people use a large amount of user-item interaction data collected from the MOOC platform to predict the dropout behaviors for further analysis. Dynamic embedding representation learning presents an attractive opportunity to model the dynamic evolution of users and items, where each user (item) can be embedded in a Euclidean space. This article introduces and analyzes the application of the joint dynamic user-item embedding algorithm in the MOOC dropout prediction. The empirical results indicated that the model has low dependence on data volume. Moreover, the model is robust to label-flipping attacks. Therefore, we believe that the model performances under different settings can be used to guide the real-world MOOC dropout prediction.

CCS CONCEPTS

• **Information systems**; • **Information systems applications**; • **Data mining**;

KEYWORDS

Dropout prediction, Dynamic embedding, Representation learning

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487141

ACM Reference Format:
Lin Wang, Zhengfei Yu, Mengru Wang, Xixi Zhu, and Yun Zhou*. 2021. MOOC Dropout Prediction Based on Dynamic Embedding Representation Learning. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487075.3487141

1 INTRODUCTION

Massive open online courses (MOOCs) have developed rapidly since they were deployed in 2006 [1]. MOOCs break the limitations of traditional courses in the classroom, and users from all over the world can learn through the Internet anytime, anywhere, which promotes the openness and sharing of education. With MOOCs, students have greater flexibility to take courses, thus attracting more and more students and allowing the platform itself to grow rapidly.

The world's top universities, such as Stanford University and Harvard University, have released MOOCs for free learning by users all over the world [2]. Stanford University launched its Introduction of Artificial Intelligence MOOC in the fall of 2011 with more than 160,000 registered users. In China, many well-known universities, such as Tsinghua University and Peking University, have launched their own MOOCs in cooperation with edX and Coursera, and the high-quality Chinese courses have won the favor of users all over the world.

Despite the rapid growth and success of MOOCs, the online courses also have their own problems. One glaring problem is the high drop-out rate. Compared with the offline teaching mode, most students who take online courses drop out of the course midway, only a small number of users can actually complete the relevant courses, and many users even become "zombie users" after registering on the platform. Statistics show that even the world's top universities - Stanford university, Massachusetts Institute of Technology and The University of California, Berkeley – also suffer

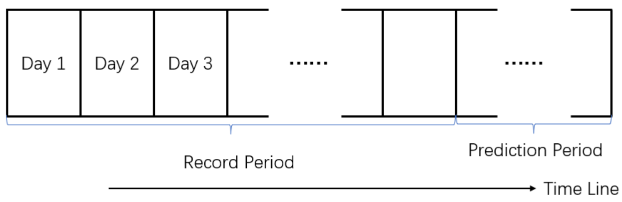

Figure 1: Illustration of MOOC Dropout Prediction.

a high drop-out rate, that reaches up to 90% [1]. A high drop-out rate is a potential factor hindering the development of MOOC platforms. Thus, effectively predicting whether a student will drop out or not is very important for MOOC platforms to make further interventions.

At present, the definition of dropout prediction is not unified. According to different data sets and prediction purposes, the specific definitions of dropout are also different, and these definitions are introduced in detail in relevant review literature [3, 4]. As shown in Figure 1, this paper formalizes the dropout prediction problem as follows: predicts whether these students will interact with the course in the future according to the interaction records of given users for some courses in a certain period of time. Therefore, we can convert it to a binary classification problem. During the forecast period, if a user has activity records for a course, it is considered that the user has not quit the corresponding course learning. Otherwise, we assume that the user has dropped out of the course.

The MOOCs platform has a large amount of time-series user-course interaction data, which contain various state information of users during their study. As Figure 2 shows, these interactions form a network of sequential interactions between the user and the courses. Each interaction transaction has the time stamp t_r and the time spent watching the video and other relevant feature vectors f_r.

Figure 2 propose a variety of algorithms to use these data to build a prediction model for users, and then identify users at risk of dropping out of classes. By modifying the relevant syllabus and content of the course, the drop-out rate can be reduced and participation can be increased [5].

In all prediction methods, representation learning, that is, the low-dimensional embedding of learning entities, provides a powerful way to represent the evolution of user and course attributes [6, 7]. Using representation learning to generate dynamic embeddings of user-course entities can support downstream tasks such as user course dropout prediction.

There are several challenges in generating dynamic embedding of user-course interaction networks using presentation learning: (1) How to accurately predict the evolution trajectory of user's dynamic embedding over time. Most existing approaches [8, 9] generate embedding only when the user has interactive behavior. However, user intentions change over time, so user's embedding needs to be updated over time. (2) How to consider both the static and dynamic attributes of entities in a unified framework. Entities have both static and dynamic properties that do not change over time. Most existing approaches [9, 10] consider only one of the two when generating embeddings. (3) How to extend to large-scale data training while maintaining the timing characteristics between

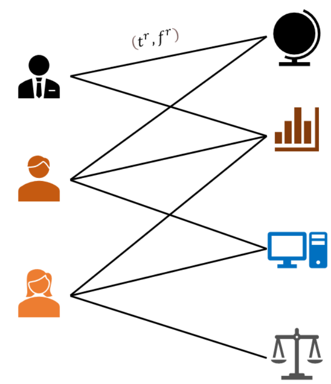

Figure 2: A temporal Interaction Network of User-Item.

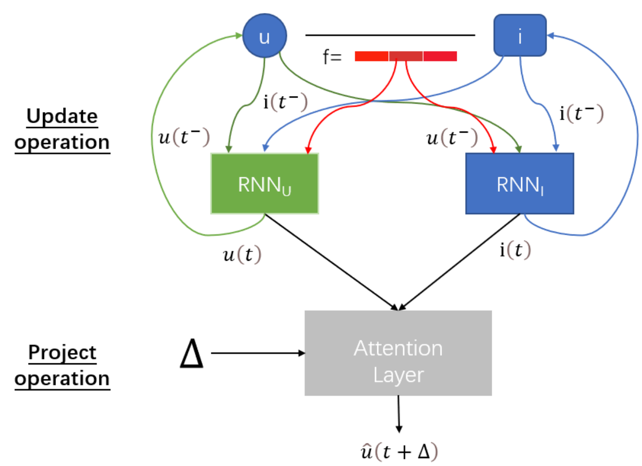

Figure 3: The JODIE Model Framework.

interactions. Most models [9, 10] are trained one interaction at a time in chronological order and are difficult to extend to large data sets.

In 2019, Kumar et al. [11] propose a new model named JODIE (Joint Dynamic User-Item Embeddings), which can generate the embedding trajectory of users and projects according to the time interaction information, and output the embedding information of entities. JODIE model provides the possibility to solve problems mentioned above. As shown in Figure 3, the JODIE model consists of two types of operations - update and projection. The update operation composed of two coupled recursive neural networks will update the embedding of users and items when there is interaction. At the same time, the projection operation composed of temporal attention layer is used to predict the passage of user embedding along with the completion time of prior interaction Δ, and the embedding trajectory of the entity is obtained. In addition, the JODIE model uses the Batch processing algorithm t-Batch to create independent interactive training batches to train the model, so that the interactions in each Batch can be processed in parallel, which means it can be extended to large-scale data sets.

Though JODIE is a general model that can be used to learn dynamic presentation learning involving multiple datasets such as Reddit, Wikipedia, LastFM and MOOCs, we focus on the dropout prediction problem of MOOCs in this paper. It should be noted that in addition to the universality of dynamic presentation learning, there are some particularities and it's not appropriate to use JOIDE directly to solve the problem. On the basis of relevant research, this paper first analyzes the application of JODIE in the prediction problem of MOOCs specifically, and then conducts a series of experiments. The main contributions of this paper include three aspects:

- Detailed analysis and introduction are made on the application of JODIE model proposed in literature [11] in MOOC dropout prediction.
- Experiments are conducted on different levels of user course interactive data, and the results show that JODIE model has low data volume dependence on MOOCs withdrawal prediction
- Experiments show that JODIE is robust to label-flipping attacks, with specific attacks of user-course interactive data.

2 RELATED WORK

2.1 MOOC Dropout Prediction Algorithm

In the past, researchers mainly used conventional machine learning algorithms such as support vector machine [12] and logistic regression [13] to predict the dropouts of users. Kloft M [12] combined the hidden Markov model with the support vector machine, and Xing W [14] combined the Bayesian network with the decision tree. Unlike other studies, Chanchary FH [15] used K-means to perform cluster analysis on the students in the MOOC platform to automatically discover inactive users.

In recent years, deep learning has also been used to predict dropouts. Fei M [16] converted the dropout prediction problem into a time-series prediction problem, and made use of a recurrent neural network (RNN) with long short-term memory (LSTM) unit to make the prediction. Wang W [17] made prediction by combining convolutional neural network and cyclic neural network, and the constructed model can automatically extract features from the original data. Qiu J [18] combined the statistical information, forum behavior data and learning behavior of users, and proposed to use the implicit dynamic factor model to predict the learning influence of users.

Prenkaj, B [19] gave a comprehensive overview of the problem of student dropout prediction in their tutorial, and Prenkaj B [3] conducted an in-depth analysis of machine learning algorithms proposed for student dropout prediction in online courses. Please refer to the tutorial and survey above for more information.

2.2 Dynamic Embedding Represents Learning

Representation learning can be regarded as a dimensionality reduction method, which maps each node to a low-dimensional vector space, so it can be well applied to various downstream tasks such as classification, prediction, regression, etc.

Several models have recently been proposed to generate embedding representations for nodes (users and items) in temporal networks. CTDNE(Continuous-Time Dynamic Network Embeddings) [20] used random walks that increase over time to generate embeddings. Similarly, IgE (Interaction Graphs Embeddings) [21] generated the final embeddings of users and items from the interaction diagram. However, because the CTDNE and IgE models generate the final static embedding of the network, the model needs to be rerun for the new edges to create the dynamic embedding. The DynamicTriad [22] algorithm supports dynamic embedded learning, but due to the model's dependence on triples, it cannot run on the two-part interactive network. Other recent algorithms, such as DDNE (Deep Dynamic Network Embedding) [23], DANE (Dynamic Attributed Network Embedding) [24], DynGem [25], DySAT (Deep Embedding Method for Dynamic Graphs) [26] and Literature [27], learn dynamic embedding from sequences of graphic snapshots, but do not apply to user-course continuous interactive data in MOOCs. Sun, L [28] proposed to learn dynamic graph representation in hyperbolic space, and introduced Temporal GNN (TGNN) based on a theoretically grounded time encoding approach. Recent models, such as NPGLM (Non-Parametric Generalized Linear Model) [29], DGNN (Dynamic Graph Neural Network) [30] and TVAE (Temporal network embedding method based on the VAE framework) [31], have embedded representations of links between learning nodes for a long time. Due to the transient nature of side interactions in temporal interaction networks, these models are not applicable to interaction networks.

3 THE PROPOSED FRAMEWORK

JODIE is a method used to learn embedding trajectories of user $u(t) \in \mathbb{R}^n,\ \forall u \in \mathcal{U}$ and item $i(t) \in \mathbb{R}^n,\ \forall i \in \mathcal{I}, \forall t \in [0, T]$ from the temporal sequence of user-item interaction $S_r = (u_r, i_r, t_r, f_r)$, where user $u_r \in \mathcal{U}$ and item $i_r \in \mathcal{I}$ interaction within time $t_r \in \mathbb{R}^+, 0 < t_1 \leq t_2 \ldots \leq T$, and the associated feature vector is f_r. For ease of representation, the subscript r of the formula used in this section will not be shown.

The framework of the proposed model JODIE is shown in Figure 3. Firstly, JODIE use the trained embedding update operation to update the embedding representation of users and items. Then, the future embedding of users can be predicted using the previously observed state and the lasting time via the embedding projection operation. Finally, the embedding of the entity are updated again, when the next interaction of users and items occurs.

3.1 Embedding Update Operation

As shown in Figure 2, for interaction $S = (u, i, t, f)$ between user u and course item i, we can yield dynamic embedding $u(t)$ for user u and $i(t)$ for course i at time t by the embedding update operation. The JODIE model is updated using two recurrent neural networks, where RNN_U is shared by users and is used to update the user embedding, and RNN_I is shared by courses and is used to update the item embedding. The RNNs, corresponding to the user and course respectively, are mutually-recursive. When user u interacts with course i, RNN_U use the embedding $i(t^-)$ of course i right before time t as input to update the embedding $u(t)$. $i(t^-)$ is the same as the embedding of item i after a previous interaction with any user.

Note that, the JODIE model, proposed to solve the dropout prediction problem of MOOCs, has static and dynamic embedding when coding users and courses entities. The static embedding $\bar{\boldsymbol{u}} \in \mathbb{R}^d$, $\forall u \in \mathcal{U}$ and $\bar{\boldsymbol{i}} \in \mathbb{R}^d, \forall i \in \mathcal{I}$ don't change over time and can be used to express fixed attributes such as the user's long-term interest. The JODIE model uses one-hot vector to code static embedding for all users and courses. At the same time, user u and course i also have dynamic embedding $u(t) \in \mathbb{R}^n$ and $i(t) \in \mathbb{R}^n$ at time t respectively, which will change over time. The dynamic embedding sequence of users/courses is called trajectory. Dynamic embedding of an entity can reflect its current state, making it more meaningful. For the same reason, RNN_I uses the embedding $u(t^-)$ of user u just before time t as input to updates the dynamic embedding $i(t)$ of item i. This results in mutually-recursive dependency between the embeddings. In other words, the update of user and course can be formalized as:

$$\boldsymbol{u}(\boldsymbol{t}) = \sigma\left(W_1^u \boldsymbol{u}(\boldsymbol{t}^-) + W_2^u \boldsymbol{i}(\boldsymbol{t}^-) + W_3^u \boldsymbol{f} + W_4^u \Delta_u\right) \tag{1}$$

$$\boldsymbol{i}(\boldsymbol{t}) = \sigma\left(W_1^i \boldsymbol{i}(\boldsymbol{t}^-) + W_2^i \boldsymbol{u}(\boldsymbol{t}^-) + W_3^i \boldsymbol{f} + W_4^i \Delta_i\right) \tag{2}$$

where Δ_u represents the time elapsed since user u interacted with any course. Δ_i represents the time elapsed since course i interacted with any user. $\boldsymbol{f}$ represents feature vector of the interaction. The matrix $W_1^u, \ldots W_4^u$ and $W_1^i, \ldots W_4^i$ are the parameter of RNN_U and RNN_I respectively. σrepresents the nonlinear sigmoid function.

3.2 Embedding Projection Operation

The JODIE model uses the embedding projection operator to predict the future embedding trajectory of users. The projection is serialized to obtain embedding trajectories, and can then be used for downstream tasks. Specifically, within the short duration Δ_1 after time t, the user's projected embedding $\hat{u}(t + \Delta_1)$ is close to its previously observed embedding $u(t)$. Given $\Delta > \Delta_2 > \Delta_1$, as time goes on, the projected embedding drift towards $\hat{\boldsymbol{u}}(\boldsymbol{t} + \Delta_2)$ and even $\hat{\boldsymbol{u}}(\boldsymbol{t} + \Delta)$. Assuming the next interaction occures at time $t + \Delta$, the algorithm updates the user's dynamic embedding to $u(t + \Delta)$ using the embedding update operation.

The embedding projection operation takes the embedding of user u at time t and the duration time Δ as two inputs. Referred to literature [8], time is considered into projected embedding through Hadamard product. It's obvious that using neural networks directly to model the interaction between the spliced inputs is inefficiency. Therefore, it is not feasible to simply concatenate the embedding and the time. Instead, the JODIE model propose to use a temporal attention vector. Specifically, JODIE model converts Δ to time-dependent vector $\boldsymbol{w} = W_p\Delta$ using linear layer W_p, and initialize W_p using zero-mean Gaussian. After that, the projected embedding is obtained as an element-wise product of the time correlation vector $\boldsymbol{w}$ with the previous embedding. We describe the process as follows:

$$\hat{u}(t + \Delta) = (1 + w) * u(t) \tag{3}$$

where the vector $1 + \boldsymbol{w}$ acts as the time attention vector to scale the user embedding $u(t)$. When Δ =0 and w=0, the projection embedding is the same as the input embedding vector. The projected embedding vector drifts over time. That is, as the value of Δ increases, the difference between the projected embedding vector and the input embedding vector becomes larger.

3.3 Training Model

The JODIE model directly outputs the course embedding vector $\tilde{i}(t + \Delta)$, instead of the interaction probability between user u and course i, which has the advantage of reducing the computation time at inference time from linear to approximate constant. Most existing methods [8, 9] take interaction probability as output, and use forward transfer operation for each course to find course with the highest probability score. By contrast, the JODIE model only does forward pass to the prediction layer once, and directly output the predicted item embedding. Then, the Locality Sensitive Hashing (LSH) [32] is used to return the item with the closest embedding in approximately constant time. In oreder to maintain the LSH data structure, the model updates the LSH when the course embedding updates.

For each interaction, the JODIE model can be trained by minimizing the L_2 distance between the predicted item embedding and the real item embedding. Finnaly, the total training loss is:

$$\text{Loss} = \sum_{(u,j,t,f)\in S} \tilde{j}(t) - \left[\bar{j}, j(t^-)\right]_2 + \lambda_U u(t) - \quad u(t^-)_2 + \lambda_I j(t) - j(t^-)_2 \tag{4}$$

The first item of the loss minimizes the error of predictive embedding. The last two items are used to regularize the loss and prevent the continuous dynamic embedding of users and courses from changing too much. λ_U and λ_I are scaling parameters to ensure that the loss is within the same range.

4 EXPERIMENT

Section 4.1 introduces the data sets used in the assessment, Section 4.2 introduces the relevant experimental setup, and Section 4.3 conducts some experiments on the JODIE model for the MOOC dropout prediction task and compares it with some other models.

4.1 Dataset

In this paper, the prediction data set of online students' drop-out behavior published by KDD Cup 2015 was used to conduct correlation analysis and experiment. The public data set consists of the interactive behaviors of students learning courses on the online platform. The dataset includes 7046 student users, 96 course items, and the resulting 411,749 interactions. The recording period for each course is 30 days. Each piece of data contains user node ID, course node ID, time stamp of interaction behavior, four dimensions characteristics, and label information indicating whether to drop the course. Among them, the characteristic dimensions mainly include "watching video time" and "browsing mode".

4.2 Implementation Details

Anticipate goals. For KDD Cup 2015, the goal is to predict whether the user-course status will change over the next 10 days, that is, whether the user will drop out of an elective course. If the user drops the course, the label "1" is assigned. For users who have not dropped out, the label is always "0". The data set recorded 4,066 dropouts, accounting for about 0.98%. The user -course interaction data in the MOOC platform belongs to the category imbalance data. Due to the skewed distribution of labels, accuracy is not an

Table 1: User Dropout Prediction Experiment

Method	KDD Cup 2015	Improvement of JODIE
Logistic Regression	0.526	18.8%
GRU	0.629	8.5%
JODIE	**0.714**	-

appropriate evaluation indicator. We use Area Under Receiver Operating Characteristic Curve (AUC) value as indicators to evaluate the performance of JODIE, which is a standard metric in tasks with highly imbalanced datasets. It should be noted that the larger the AUC value, the stronger the generalization ability of the model.

Partition of data. During the experiment, the data was segmented in chronological order, the model was trained on the first 80% of interaction data, verified on the subsequent 10% of interaction data, and tested on the last 10% of interaction data. Thus, the model is trained to simulate the actual interaction. 128 is used as the dynamic embedding dimension of all algorithms, and the one-hot vectors are used as the static embedding. For fair comparison, all the algorithms were run for 20 rounds, and the experimental results were taken from the test data with the best performance on the validation set.

Baseline algorithm. Logical regression and Gate Recurrent Unit(GRU) are used as baseline algorithms in this paper.

Experiment 1: Prediction of dropout of MOOC platform users.

The experimental results of Jodie and baseline model in the prediction of users of MOOCs platform dropping out of classes are shown in Table 1. As can be seen from the table, the AUC obtained by the JODIE is higher than the two baseline models. It is 18.8% and 8.5% higher than logistic regression algorithm and recurrent neural network, respectively.

Experiment 2: Accuracy of prediction for data of different magnitude

In this experiment, we verify the accuracy of JODIE in the dropout prediction problem of MOOCs by changing the magnitude of data. Here, we take the first 50,000 to 300,000 items from the complete data set as a sub-data set according to the time sequence of the interactions and on an increasing scale of 50,000 items. A total of 6 data sets can be obtained. In each case, the first 80 percent of interactions are used as training, the next 10 percent as validation, and the next 10 percent as testing. Figure 4 shows the AUC of all algorithms on MOOC data of different magnitude. We noticed that JODIE's predicted performance was stable as the magnitude of the data changed and did not vary much from one data point to another, consistently outperforming the baseline model. We suggest that the stability of JODIE's performance may come from the attention layer of the model, which enables the model to assign greater weight to more recent interactions.

Experiment 3: Robustness to different degrees of label flip attacks.

Finally, we verify the effect of different degree of label flip attacks [33] on model prediction. We distinguish the training data according to the positive and negative examples. According to the scale of 1%-5%, the data labels in the training data are flipped randomly

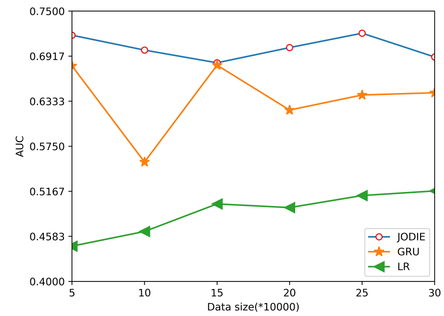

Figure 4: AUC of Predictions On Different Levels Of Data Size.

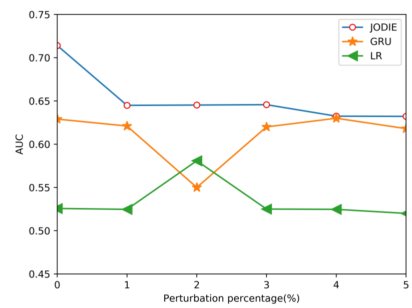

Figure 5: Robustness to Different Degrees of Label-Flip Attacks.

respectively to make the positive examples become negative ones and vice versa. For the data set after the label flip attack, the model is used to predict the dropout, so as to evaluate the robustness of the model. As shown in Figure 5, we found that the label flip attack has little impact on the performance of JODIE, and the overall performance is the best.

5 CONCLUSIONS

Prediction of user drop-out is an important task in MOOCs platform. This paper analyzes the performance of JODIE model for this task, and gives a detailed description and introduction of the model's framework. Moreover, the performance of JODIE model and baseline algorithm is compared through a series of experiments. The results show that, compared with the baseline algorithms, JODIE model can perform better in the MOOC dropout prediction task, maintain higher accuracy for different levels of data, and maintain better robustness when subjected to different degrees of tag flip attacks. This article demonstrates that JODIE has a certain practical significance in this application, even learning with noisy training data. More complex situations could be considered in the future work.

ACKNOWLEDGMENTS

This work is supported by Research Project of Teaching Reform in Hunan province ([2018]436) and Training Program for Excellent Young Innovators of Changsha (KQ2009009).

REFERENCES

[1] Daniel J(2012). Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility[J]. Journal of interactive media in education, 2012:257-284.

[2] Feng W, Tang J, Liu T X (2019). Understanding dropouts in MOOCs[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 33(01): 517-524.

[3] Prenkaj B, Velardi P, Stilo G, *et al* (2020). A Survey of Machine Learning Approaches for Student Dropout Prediction in Online Courses[J]. ACM Computer Surveys, 53(3): Article 57.

[4] Gardner J, Brooks C (2018). Student success prediction in MOOCs[J]. User Modeling and User-Adapted Interaction, 28(2):127-203.

[5] Cui Y, Chen F, Shiri A, *et al* (2019). Predictive analytic models of student success in higher education: A review of methodology[C].

[6] Tirumala SS (2020). Evolving deep neural networks using coevolutionary algorithms with multi-population strategy[J]. Neural Computing and Applications, 32(16):13051-13064.

[7] Farajtabar M, Gomez-Rodriguez M, Wang Y, *et al* (2018). COEVOLVE: A Joint Point Process Model for Information Diffusion and Network Co-evolution[C]. 27th International World Wide Web, WWW 2018, April 23, 2018 - April 27, 2018: 473-477.

[8] Beutel A, Covington P, Jain S, *et al* (2018). Latent Cross: Making Use of Context in Recurrent Recommender Systems[C]. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 46–54.

[9] Dai H, Wang Y, Trivedi R, *et al* (2016). Deep coevolutionary network: Embedding user and item features for recommendation[J]. arXiv preprint arXiv:1609.03675.

[10] Wang Y, Du N, Trivedi R, *et al* (2016). Coevolutionary latent feature processes for continuous-time user-item interactions[C]. 30th Annual Conference on Neural Information Processing Systems, NIPS 2016, December 5, 2016 - December 10, 2016:4554-4562.

[11] Kumar S, Zhang X, Leskovec J (2019). Predicting dynamic embedding trajectory in temporal interaction networks[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1269-1278.

[12] Kloft M, Stiehler F, Zheng Z, *et al* (2014). Predicting MOOC Dropout over Weeks Using Machine Learning Methods[C]. EMNLP 2014.

[13] Taylor C, Veeramachaneni K, O Reilly U (2014). Likely to stop? Predicting Stopout in Massive Open Online Courses[J]. ArXiv, abs/1408.3382.

[14] Xing W, Chen X, Stein J, *et al* (2016). Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization[J]. Computers in Human Behavior, 58:119-129.

[15] Chanchary FH, Haque I, Khalid MS (2008). Web usage mining to evaluate the transfer of learning in a web-based learning environment[C]. 1st International Workshop on Knowledge Discovery and Data Mining, WKDD, January 23, 2008 - January 24, 2008:249-253.

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

[17] Wang W, Yu H, Miao C (2017). Deep Model for Dropout Prediction in MOOCs[C]. Proceedings of the 2nd International Conference on Crowd Science and Engineering, Beijing, China, 26–32.

[18] Qiu J, Tang J, Liu TX, *et al* (2016). Modeling and Predicting Learning Behavior in MOOCs[C]. Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, California, USA, 93–102.

[19] Prenkaj B, Stilo G, Madeddu L (2020). Challenges and solutions to the student dropout prediction problem in online courses[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 3513-3514.

[20] Nguyen GH, Lee JB, Rossi RA, *et al* (2018). Continuous-Time Dynamic Network Embeddings[C]. Companion Proceedings of the The Web Conference 2018, Lyon, France, 969–976.

[21] Zhang Y, Xiong Y, Kong X, *et al* (2017). Learning node embeddings in interaction graphs[C]. 26th ACM International Conference on Information and Knowledge Management, CIKM 2017, November 6, 2017 - November 10, 2017:397-406.

[22] Zhou L, Yang Y, Ren X, *et al* (2018). Dynamic network embedding by modeling triadic closure process[C]. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, February 2, 2018 - February 7, 2018:571-578.

[23] Li, T., *et al*(2018). Deep Dynamic Network Embedding for Link Prediction. IEEE Access,. 6: p. 29219-29230.

[24] Li, J., *et al* (2017). Attributed network embedding for learning in a dynamic environment. in 26th ACM International Conference on Information and Knowledge Management, CIKM 2017, November 6, 2017 - November 10. 2017. Singapore, Singapore: Association for Computing Machinery.

[25] Goyal, P., *et al* (2018). DynGEM: Deep Embedding Method for Dynamic Graphs. ArXiv. abs/1805.11273.

[26] Sankar, A., *et al* (2020). DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks, in Proceedings of the 13th International Conference on Web Search and Data Mining. Association for Computing Machinery: Houston, TX, USA. p. 519–527.

[27] Rahman, M., *et al* (2018). DyLink2Vec: Effective Feature Representation for Link Prediction in Dynamic Networks. ArXiv. abs/1804.05755.

[28] Sun, L., *et al* (2021). Hyperbolic Variational Graph Neural Network for Modeling Dynamic Graphs, in AAAI 2021. p. 4375-4383.

[29] Sajadmanesh S, Bazargani S, Zhang J, *et al* (2019). Continuous-time relationship prediction in dynamic heterogeneous information networks[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 13(4): 1-31.

[30] Ma, Y., *et al.*, Dynamic Graph Neural Networks (2018). ArXiv. abs/1810.10627.

[31] Jiao, P., *et al* (2021). Temporal Network Embedding for Link Prediction via VAE Joint Attention Mechanism. IEEE Transactions on Neural Networks and Learning Systems, p. 1-14.

[32] Leskovec, J., A. Rajaraman, and J.D. Ullman (2014). Mining of massive datasets: Second edition. Mining of Massive Datasets: Second Edition: Cambridge University Press. 1-458.

[33] Yu Zheng-Fei, Yan Qiao, Zhou Yun (2021). A survey on adversarial machine learning for cyberspace defense. Acta Automatica Sinica, 47(x): 1−25. doi: 10.16383/j.aas.c210089