Temporal Graph-based CNNs (TG-CNNs) for Online Course Dropout Prediction

Zoe Hancox*[0000−0003−0473−5971] and Samuel D. Relton[0000−0003−0634−4587]

The University of Leeds, Leeds, UK
Z.L.Hancox@leeds.ac.uk, S.D.Relton@leeds.ac.uk

Abstract. Due to the global pandemic, the use of online courses is increasing significantly; yet the rate of student dropout from online courses is rising. The Accessible Culture & Training Massive Open Online Course (ACT MOOC) dataset is comprised of a temporal sequence of student actions and subsequent dropout information. We introduce a novel approach based upon temporal graphs, which uses the sequence of (and time between) events to predict dropout. The dataset consists of 7,047 users, with a dropout rate of 57.7%. The Temporal Graph-Based Convolutional Neural Network (TG-CNN) models developed in this study are compared against baseline models and existing models in the literature. Performance is assessed using the AUC, accuracy, precision, recall, and F1 score. Our novel TG-CNN model achieves an AUC score of **0.797**, which improves upon previous literature: JODIE 0.756, TGN + MeTA 0.794, TGN 0.777, and CoPE 0.762. Our model offers a novel and intuitive formulation of this problem, with state-of-the-art performance.

Keywords: Temporal Graphs · Dropout Prediction · Neural Networks

1 Introduction

Massive Open Online Courses (MOOCs) allow people to study and learn a wide range of material wherever and whenever they choose [8]. Despite this, student retention with MOOCs is low and course dropout is high [15]. The COVID-19 pandemic has caused a rise in the number of students partaking in online courses. Simultaneously, MOOC dropout rates are increasing as educational resources are forced to move online and an epidemic of screen fatigue sets in [13]. Additionally, it has been noted that dropout rates are higher from MOOCs compared to in-person and off-line courses [13]. Predicting user dropout based on clickstream data could enable identification of behaviour patterns prior to dropout, to target interventions designed to encourage course completion [7, 5].

Graph networks (GNs) are rising in popularity in machine learning (ML) [12, 16]. GNs capture object interactions, and are used to represent social networks and recommender systems, for example where nodes may represent people and edges depict messages from one person to another. Convolutions applied over graph structures have been shown to learn effectively in various tasks [12, 1].

* Corresponding author

The model described in this paper utilises temporal GNs, a three-dimensional Convolutional Neural Network (3D CNN) and Long Short-Term Memory (LSTM) units, to evaluate clickstream data (user actions) making use of the elapsed time between events to improve predictive performance of student dropout.

The 3D CNN component has the ability to capture short-term temporal patterns of user actions, whilst the LSTM can cover longer-term associations. Our formulation of this problem as a graph classification task diverges from previous work in this area, where node and edge level classification are typically used. The approach outlined in this research utilises temporal graphs and 3D CNNs to represent, and learn from, structured temporal data. This generates two novel contributions:

1. Previous MOOC articles have focused on classifying the nodes/edges within a graph, as opposed to our formulation which classifies entire graph structures.
2. LSTMs assume a constant elapsed time between sequence elements, an issue which has received some attention in the literature [4, 15]. Our TG-CNN approach offers an alternative formulation of this problem (including variable time dilation), which can model more complex temporal links.

2 Methods

2.1 Dataset and Modelling Approach

The Accessible Culture & Training MOOC (ACT MOOC) dataset includes timestamped actions and dropout labels for 7,047 users[1]. There are 97 potential clickstream actions a user can take which are timestamped. Timestamps are counted in seconds from the first interaction a user makes with the online course. In total over 411,749 interactions are captured, with the highest number of actions taken by one user totalling 505. Dropout occurred in 57.69% of users.

Our approach to this dropout prediction task is to turn this sequence of events into a temporal multigraph and formulate dropout prediction as a binary graph classification task, where each student has their own temporal graph to be classified. In particular, the $n = 97$ possible actions form the nodes of this graph and the temporal edges capture the elapsed time between actions. This can be stored in a 3D tensor $G(i, j, k) = t_k$ where $i, j \in \{1, \ldots, n\}$ are nodes in the graph, and t_k is the elapsed time for the kth edge in the temporal graph. An example with 4 possible actions is shown in Figure 1.

In practice, we actually store $G(i, j, k) = \exp(-\gamma t_k)$, where $\gamma > 0$ is a trainable parameter of the model. This has two benefits:

1. Actions taken in quick succession or simultaneously have a value close to 1 and actions with a greater temporal gap are closer to 0. Events that are not related have value zero. This allows the temporal graph to be stored as a sparse 3-tensor, saving significant memory in the representation of the data.

[1] Stanford Network Analysis Project - https://snap.stanford.edu/data/act-mooc.html

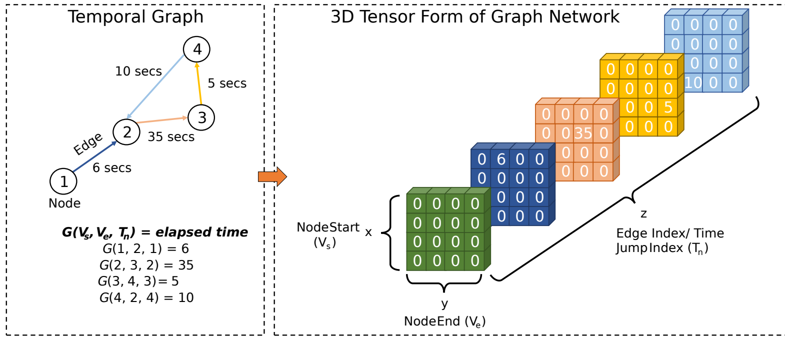

Fig. 1. Graph network visualisation showing connections between actions completed by a user in both graph form and tensor form. This example has only 4 possible actions, so is much smaller than the $97 \times 97 \times 100$ tensor that is used in this project.

2. Elapsed time can be rescaled to avoid extreme values in the neural network and potential under/overflow when using half-precision arithmetic.

The classification of these temporal graphs is performed using a neural network architecture based upon 3D convolutions. In particular, each convolution covers $m = 2, 3, 4, \ldots$ consecutive timesteps and each filter, of size $n \times n \times m$, is applied across the temporal dimension of the 3-tensor. The filters utilise information about the sequence of actions taken and the elapsed time between actions and the convolution operation (with stride 1) collapses this 3-tensor of size $n \times n \times K$ into a vector of length $K - m + 1$. The output of the convolutions is a sequence of feature vectors capturing short patterns (accounting for elapsed time); we pass these to an LSTM which allows for longer temporal patterns, built from these sub-patterns, to be captured.

For this particular task we use the most recent 100 actions of each user to reduce computational burden, front-padding any 3-tensors representing sequences of length less than 100 to ensure the most recent actions are always at the end of the 3-tensor. We also experiment with using a secondary pipeline of filters with stride 2, referred to as a "2-stream architecture", and concatenate the two before the final FCLs of the neural network.

Torch version 1.7.0, Tensorflow 2.8.0, NumPy 1.19.2, Pandas 1.2.4, Scikit-Learn 0.23.1, and CUDA 10.2.89 were used on a desktop with a NVIDIA RTX 3090 (Table 1), and the N8 Bede machine based at Durham University: an IBM Power 9 system with NVIDIA V100 GPUs (Table 2).

2.2 Model Architecture

Our proposed Temporal Graph-based Convolutional Neural Network (TG-CNN) model is shown in Figure 2. The model described can handle data that is irregularly sampled in time. The input 3-tensor of size $97 \times 97 \times 100$ is fed into the 3D

CNN layer, extracting information on the sequence of actions and elapsed times. This is then passed through a Batch Normalisation function and a Rectified Linear unit (ReLU) activation function, before proceeding through the LSTM. The output of the LSTM has dropout applied, passing the hidden features into a FCL. Dropout and a ReLU are then used again before a final FCL. Binary cross entropy logits loss is used for the binary target. Adam optimisation with L2 regularization is used to smooth oscillations during training. Our implementation also utilises a learning rate (LR) scheduler, multiplying the LR by 0.9 with an exponential decay each 10,000 steps. Early stopping is used with a patience of 50, to checkpoint the model when the validation loss decreases, interrupting execution when the model gets stuck in a local minima.

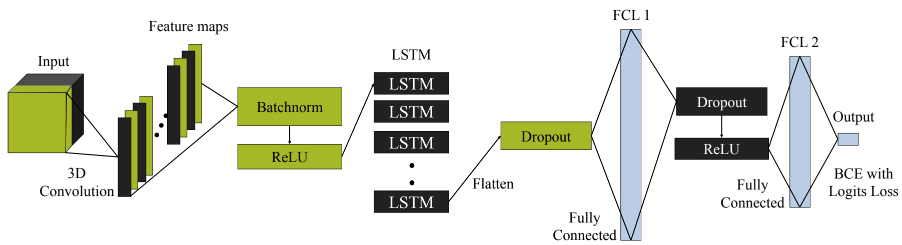

Fig. 2. Model architecture of proposed TG-CNN.

The γ variable controlling time dilation is a trainable parameter within the model. Longer gaps between actions could mean the action was less relevant, which could be modelled by increasing γ. An ablation study was performed to test the model with the γ variable, without the LSTM layer, without the exponential scaling parameter and with elastic net (L1 + L2) regularisation.

Additionally, we investigate the impact of adding a second stream (2-stream) to the network, where there is a second row 3D-CNN and LSTM using filters with a stride of 2, whereas the original 1-stream has a stride of 1. The output of the two independent streams are concatenated after the FCLs.

This model has interpretability potential, as the CNN features could be viewed directly and visualised to show the filters learnt from the data, whereas with LSTMs this is more difficult to comprehend.

2.3 Model Evaluation

This work adopts a 80/10/10 train/validation/test split for the TG-CNN models. Previous models using similar data have primarily focused on area under the receiver operating characteristic curve (AUC), so hyperparameters were optimised for best AUC score on the validation set and test set results are reported.

To optimise the model based the on validation set AUC value, we conducted a random hyperparameter search by sampling the number of epochs [25, 50, 75, 100], LR [0.1, 0.01, 0.05, 0.001, 0.005, 0.0001], number of filters [32, 64, 128], filter size [4, 16, 32, 64], number of LSTM hidden cells [16, 32, 64, 128, 256], L2 regularisation (L2 reg) parameter [1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 5e-2, 5e-3, 5e-4], FCL size [128, 256, 512, 1028, 2056], and dropout rate [0.2, 0.3, 0.4, 0.5]. For the 2-stream models, the two streams each had filters of the same size but with different strides. This results in 230,400 possible combinations of hyperparameter values, necessitating the use of random search instead of a grid search.

We also fit two baseline models to the dataset for comparison, an LSTM (BL-LSTM) and Recurrent Neural Network (BL-RNN) architecture - a single RNN layer (LSTM or RNN respectively) followed by two FCLs. Five-fold cross-validation was used to optimise the AUC over the hyperparameter combinations for these baseline models. The BL-LSTM and BL-RNN models were tuned by optimising the LR, the number of epochs, the hidden units in the RNNs, and the number of hidden neurons in the FCLs.

3 Results

We tested the TG-CNN models with 1,430 hyperparameter samples from the list in Section 2.3. The best performing hyperparameters and performance metrics for these models are shown in Table 1, metrics are averaged over ten runs to show robustness. BL-LSTM and BL-RNN were each fitted using 5-fold cross-validation with 576 different hyperparameter combinations. The best performing BL-LSTM achieved an AUC of 0.783, 79.26% accuracy, 0.800 precision, 0.852 recall and an F1-score of 0.824 within 20 epochs, using a LR of 0.01, 128 hidden LSTM neurons and FCL sizes of 32 and 16. The best performing BL-RNN achieved an AUC of 0.778, 78.86% accuracy, 0.801 precision, 0.844 recall and an F1-score of 0.819 within 20 epochs, with a LR of 0.001, 32 hidden RNN neurons and FCL sizes of 64 and 32.

Table 2 shows the best variants of each model in the ablation study, and compares to existing models in the literature (which also use the model with maximal AUC for a fair comparison). The best performing variant overall was the TG-CNN with fixed time dilation $\gamma = 1$ (AUC 0.797), closely followed by the 2-stream version (AUC 0.796). The average results after 10 re-runs led to the 2-stream model achieving the best performance (Table 1). Eliminating the LSTM component led to significantly poorer results (AUC 0.705). These results show the importance of both the multi-streams and LSTM component in achieving good performance with this approach.

4 Discussion

Table 2 shows the predictive performance of models with the best AUC score using the ACT MOOC dataset. Our novel TG-CNN approach has state-of-the-art performance on this task, despite being more intuitive and conceptually simpler than some of the other approaches in previous literature.

Table 1. Hyperparameter values and test set metrics for the best performing variants of the TG-CNN architecture (mean ± standard deviation from 10 runs).

Parameter	Variable γ	1-stream	2-stream	No LSTM	No exp	Elastic Net
Epochs	50	75	52	25	100	100
LR	0.05	0.05	0.0005	0.0001	0.001	0.001
# Filters	64	64	128	64	32	32
Filter Size	16	32	4	4	64	32
RNN Cells	128	64	32	N/A	16	16
L2 Reg	1e-3	1e-2	5e-4	N/A	N/A	N/A
FCL Size	1028	512	2056	512	1024	1024
Dropout	0.5	0.5	0.3	0.5	0.3	0.5
Test AUC	0.662±0.08	0.748±0.02	**0.763±0.01**	0.705±0.02	0.710±0.02	0.711±0.02
Accuracy	0.703±0.06	0.771±0.02	**0.775±0.01**	0.583±0.01	0.690±0.02	0.702±0.02
Precision	0.687±0.06	0.764±0.02	**0.773±0.02**	0.581±0.01	0.677±0.02	0.688±0.02
Recall	0.923±0.05	0.883±0.02	0.859±0.04	**1.00±0.00**	0.871±0.02	0.886±0.02
F1-Score	0.782±0.03	**0.817±0.01**	0.811±0.02	0.735±0.01	0.761±0.01	0.774±0.02

Table 2. Best AUC results of user dropout prediction using the ACT MOOC dataset, from our results (left columns) and from the results in the literature (right columns).

TG-CNN and Baseline Models	**AUC**		**Literature Models**	**AUC**
TG-CNN 1-stream	**0.797**		TGN + MeTA [16]	0.794
TG-CNN 2-stream	0.796		TGN + TNS [17]	0.791
BL-LSTM	0.783		TGN [12]	0.777
BL-RNN	0.779		CoPE [21]	0.762
TG-CNN 1-stream $\gamma = 4.819$	0.760		JODIE [7]	0.756
TG-CNN with Elastic Net	0.758		TGAT + TNS [17]	0.755
TG-CNN without LSTM	0.750		NPPCTNE [23]	0.745
TG-CNN without the Exponential	0.744		TGAT [16]	0.743

The ablation study demonstrated that the LSTM layer and the exponential function enabled the model to learn more effectively, this is potentially due to the LSTM enabling long-term memory alongside the filters learnt from the CNN. The γ variable converged to an average value of 4.819 in the best performing model, which suggests that actions taken closer together are more important to predicting dropout than actions further apart. The cut-off caused by $\widehat{t} = \exp(-4.819 \times t)$ is sharper than when $\gamma = 1$, therefore when $\gamma = 4.819$ and the elapsed time t is more than 47 seconds $\widehat{t}$ will round to 0.

Other advantages of the TG-CNN approach include the constant tensor size, allowing for optimisation of the underlying linear algebra operations, and the ability to extract temporal features in parallel using 3D convolutions, as opposed to RNN-based architectures that require sequential processing through time. The 3-tensor structure enables the filters to be extracted back into a intuitive graph structure. This could serve as a visual tool to show which sequences of events and temporal patterns lead to dropout.

5 Related Work in MOOC Dropout

We searched the IEEE database using the terms "MOOC AND predict*". 95 papers were found, 24 of these were analysed from their title and abstract, and we found that 4 used CNNs [18, 11, 22, 20]. Only 1 paper utilised GNs [15], where the problem was formulated as a node/edge prediction task over time, to which they applied a novel data augmentation approach to existing models. This differs from our formulation of this problem as a graph-level classification task.

Learner behaviour feature matrices, weighted by importance, have been used alongside CNNs to predict dropout from clickstream data and improve predictive accuracy compared to basic models [18, 22, 8, 11]. In [20], the authors use CNNs alongside Squeeze-and-Excitation Networks (SE-Net) and a Gated Recurrent unit (GRU). The GRU enables maintenance of the time series relationship between the clickstream data and the SE-Net helps with automatic feature extraction, this resulted in an accuracy above 90% on their dataset. Interestingly, Edmond Meku Fotso et al. found simple RNNs provided better accuracy compared to LSTMs and GRUs [4]. Standard ML algorithms and ensemble methods including Support Vector Machines, Logistic Regression, Multi-layered Perceptrons, and Decision Trees have also been applied to this task [6, 8, 5].

Moving away from clickstream data, video views and quiz behaviour have been identified as significant factors contributing to dropout prediction [4]. In other work, course information and the type of interaction (solving problems v.s. watching videos etc.) were found to be important in an analysis based upon GRUs with attention weightings [10].

The JODIE model (see Table 2) utilises RNNs to learn and update embeddings that represent individual interactions between users and actions [7]. The actions and users each have their own RNNs to generate separate static and dynamic embeddings. The embeddings dynamically change over time, capturing the temporal aspect in a statically sized graph. These two RNNs are used together for the user embeddings to update the item embedding and vice versa. The JODIE model alters the embeddings significantly after longer periods of time, implicitly assuming that actions taken closer together have smaller impact.

6 Related Work in Graph Learning

Searching the Web of Science and IEEE databases using the terms "Convolution AND (3d OR three$dimension*) AND (time OR temporal) AND graph AND predict* AND network$" returned 18 papers. Of these, there were 5 relevant papers using temporal GNs [1–3, 19, 9], though they were focused on node and edge detection. To the best of our knowledge, this is the first work to develop temporal graph-based 3D CNN models for graph-level classification.

At the time of writing (12th May 2022), Kumar et al. had 200 citations of their paper [7]. To observe if any other researchers had used the ACT MOOC dataset processed by Kumar et al. (7,047 users), we screened these 200 papers and found 11 papers performing dropout/node prediction tasks.

Four of these utilised RNN components in their model architectures to process time, e.g. [14]. The others used graph models, all based on a node/edge classification formulation of the task. Wang et al. used temporal graph networks (TGNs) with dual message passing mechanisms (TGN + MeTA), to augment data and retain semantics for edge-level prediction and node classification [16]. These messaging passing techniques involve memory translation and cross-level propagation, to adapt the model with temporal and topological features to ignore noise more effectively. This increases the previously obtained AUC scores by 1.7% [12], with no cost to efficiency and reducing the overfitting that occurs due to noisy data. Other models tweak neighbourhood propagation techniques using temporal information e.g. [17, 23]. Zhang et al. use ordinary differential equations and GNs to observe model changes over time and for information propagation [21]. By contrast, our TG-CNN approach makes use of a novel 3-tensor structure, storing the temporal graphs in a sparse and intuitive format, which is easily amenable to feature extraction using convolutions for graph classification.

7 Study Limitations and Future Work

The ACT-MOOC dataset in this project provides clickstream events as numerical labels. Events/clickstreams descriptions are not provided. Therefore, reasoning for dropout cannot be explained.

Limits on the amount of computation time available meant we were unable to perform a full grid-search of the TG-CNN hyperparameters, and it is likely that a more optimal configuration could be found. Nevertheless, this approach improves upon previous work and can be adapted to a range of different graph classification tasks. In future work we aim to incorporate attention mechanisms into this approach, to enable further trust in the model and explain why certain predictions may have occurred [10].

The variant of the model including the time dilation factor γ as a trainable parameter performed the poorest (Table 1). The reason behind this is unclear, though the additional complexity modelled by the time dilation will increase the difficulty of the underlying optimisation problem; it is possible that a more optimal hyperparameter set could be found to improve this performance.

8 Conclusions

We propose a novel model for the classification of temporal graphs, using student online course dropout data to develop and test the method. Our approach provides a unique formulation of this problem, compared to previous formulations of node/edge prediction tasks. This method improves upon current state-of-the-art models in terms of AUC score, and our approach has a number of other benefits in terms of memory utilisation and parallel processing compared with other approaches. In future work we aim to extend this approach further, incorporating attention mechanisms to improve explainability of the model output.

Acknowledgements This work made use of the facilities of the N8 Centre of Excellence in Computationally Intensive Research provided and funded by the N8 research partnership and EPSRC (Grant No. EP/T022167/1). ZH is supported through funding by the EPSRC (Grant No. EP/S024336/1).

References

1. Chen, J., Liao, S., Hou, J., Wang, K., Wen, J.: GST-GCN: A Geographic-Semantic-Temporal Graph Convolutional Network for Context-aware Traffic Flow Prediction on Graph Sequences. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). vol. 2020, pp. 1604–1609. IEEE (Oct 2020). https://doi.org/10.1109/SMC42975.2020.9282828, https://ieeexplore.ieee.org/document/9282828/
2. Cheng, D., Wang, X., Zhang, Y., Zhang, L.: Graph Neural Network for Fraud Detection via Spatial-temporal Attention. IEEE Transactions on Knowledge and Data Engineering **14**(8), 1–1 (2020). https://doi.org/10.1109/TKDE.2020.3025588, https://ieeexplore.ieee.org/document/9204584/
3. Doosti, B., Naha, S., Mirbagheri, M., Crandall, D.: HOPE-Net: A Graph-based Model for Hand-Object Pose Estimation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 6607–6616 (Mar 2020), https://arxiv.org/abs/2004.00060v1
4. Edmond Meku Fotso, J., Batchakui, B., Nkambou, R., Okereke, G.: Algorithms for the Development of Deep Learning Models for Classification and Prediction of Behaviour in MOOCS. Proceedings of 2020 IEEE Learning With MOOCS, LWMOOCS 2020 pp. 180–184 (2020). https://doi.org/10.1109/LWMOOCS50143.2020.9234363
5. Haiyang, L., Wang, Z., Benachour, P., Tubman, P.: A time series classification method for behaviour-based dropout prediction. Proceedings - IEEE 18th International Conference on Advanced Learning Technologies, ICALT 2018 pp. 191–195 (2018). https://doi.org/10.1109/ICALT.2018.00052
6. Hong, B., Wei, Z., Yang, Y.: Discovering learning behavior patterns to predict dropout in MOOC. In: 2017 12th International Conference on Computer Science and Education (ICCSE). pp. 700–704. No. Iccse, IEEE (Aug 2017). https://doi.org/10.1109/ICCSE.2017.8085583, http://ieeexplore.ieee.org/document/8085583/
7. Kumar, S., Zhang, X., Leskovec, J.: Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1269–1278. ACM, New York, USA (Jul 2019). https://doi.org/10.1145/3292500.3330895, https://dl.acm.org/doi/10.1145/3292500.3330895
8. Liu, K., Tatinati, S., Khong, A.W.: A weighted feature extraction technique based on temporal accumulation of learner behavior features for early prediction of dropouts. Proceedings of 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering, TALE 2020 pp. 295–302 (2020). https://doi.org/10.1109/TALE48869.2020.9368317
9. Pham Van, L.L., Bach Tran, Q., Pham, T.L., Long Tran, Q.: Node-aware convolution in Graph Neural Networks for Predicting molecular properties. Proceedings - 2020 12th International Conference on Knowledge and Systems Engineering, KSE 2020 pp. 120–125 (Nov 2020). https://doi.org/10.1109/KSE50997.2020.9287744

10. Pulikottil, S.C., Gupta, M.: ONet - A Temporal Meta Embedding Network for MOOC Dropout Prediction. Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020 pp. 5209–5217 (2020). https://doi.org/10.1109/BigData50022.2020.9378001
11. Ren, Y., Huang, S., Zhou, Y.: Deep learning and integrated learning for predicting student's withdrawal behavior in MOOC. Proceedings - 2021 2nd International Conference on Education, Knowledge and Information Management, ICEKIM 2021 pp. 81–84 (2021). https://doi.org/10.1109/ICEKIM52309.2021.00026
12. Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.: Temporal Graph Networks for Deep Learning on Dynamic Graphs. arXiv preprint pp. 1–16 (Jun 2020), http://arxiv.org/abs/2006.10637
13. Sun, Z., Harit, A., Yu, J., Cristea, A.I., Shi, L.: A Brief Survey of Deep Learning Approaches for Learning Analytics on MOOCs, vol. 12677 LNCS. Springer International Publishing (2021). https://doi.org/10.1007/978-3-030-80421-3_4, http://dx.doi.org/10.1007/978-3-030-80421-3_4
14. Wang, A.Z., Ying, R., Li, P., Rao, N., Subbian, K., Leskovec, J.: Bipartite Dynamic Representations for Abuse Detection. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 3638–3648 (2021). https://doi.org/10.1145/3447548.3467141
15. Wang, L., Wang, H.: Learning behavior analysis and dropout rate prediction based on MOOCs data. Proceedings - 10th International Conference on Information Technology in Medicine and Education, ITME 2019 pp. 419–423 (2019). https://doi.org/10.1109/ITME.2019.00100
16. Wang, Y., Cai, Y., Liang, Y., Ding, H.: Adaptive Data Augmentation on Temporal Graphs. NeurIPS (NeurIPS), 1–13 (2021)
17. Wang, Y., Cai, Y., Liang, Y., Ding, H., Wang, C., Hooi, B.: Time-Aware Neighbor Sampling for Temporal Graph Networks. arXiv preprint pp. 1–12 (2021), http://arxiv.org/abs/2112.09845
18. Wen, Y., Tian, Y., Wen, B., Zhou, Q., Cai, G., Liu, S.: Consideration of the local correlation of learning behaviors to predict dropouts from MOOCs. Tsinghua Science and Technology **25**(3), 336–347 (2020). https://doi.org/10.26599/TST.2019.9010013
19. Wu, Z., Wang, M., Wang, J., Zhang, W., Fang, M., Xu, T.: DeepWORD: A GCN-based Approach for Owner-Member Relationship Detection in Autonomous Driving. Institute of Electrical and Electronics Engineers (IEEE) (Mar 2021), https://arxiv.org/abs/2103.16099v2
20. Zhang, Y., Chang, L., Liu, T.: MOOCs Dropout Prediction Based on Hybrid Deep Neural Network. Proceedings - 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2020 pp. 197–203 (2020). https://doi.org/10.1109/CyberC49757.2020.00039
21. Zhang, Y., Xiong, Y., Li, D., Shan, C., Ren, K., Zhu, Y.: CoPE: Modeling Continuous Propagation and Evolution on Interaction Graph. International Conference on Information and Knowledge Management, Proceedings pp. 2627–2636 (2021). https://doi.org/10.1145/3459637.3482419
22. Zheng, Y., Gao, Z., Wang, Y., Fu, Q.: MOOC Dropout Prediction Using FWTS-CNN Model Based on Fused Feature Weighting and Time Series. IEEE Access **8**, 225324–225335 (2020). https://doi.org/10.1109/ACCESS.2020.3045157
23. Zhou, Y., Luo, S., Pan, L., Liu, L., Song, D.: Continuous temporal network embedding by modeling neighborhood propagation process. Knowledge-Based Systems **239**, 107998 (2022). https://doi.org/10.1016/j.knosys.2021.107998, https://doi.org/10.1016/j.knosys.2021.107998