Intervention Prediction in MOOCs based on Learners' Comments: A Temporal Multi-Input Approach Using Deep Learning and Transformer Models

Laila Alrajhi[1], Ahmed Alamri[1] and Alexandra I. Cristea[1]

[1] Computer Science, Durham University, Durham, UK,
[1]{laila.m.alrajhi, ahmed.s.alamri, alexandra.i.cristea}@durham.ac.uk

Abstract. High learner dropout rates in MOOC-based education contexts have encouraged researchers to explore and propose different intervention models. In discussion forums, intervention is critical, not only to identify *comments* that require replies but also to consider *learners* who may require intervention in the form of staff support. There is a lack of research on the role of intervention based on learner comments to prevent learner dropout in MOOC-based settings. To fill this research gap, we propose an intervention model that detects when staff intervention is required to prevent learner dropout using a dataset from FutureLearn. Our proposed model was based on learners' comments history by integrating the most-recent sequence of comments written by learners to identify if an intervention was necessary to prevent dropout. We aimed to find both the proper classifier and the number of comments representing the appropriate most recent sequence of comments. We developed several intervention models by utilising two forms of supervised multi-input machine learning (ML) classification models (deep learning and transformer). For the transformer model, specifically, we propose the siamese and dual temporal multi-input, which we term the multi-siamese BERT and multiple BERT. We further experimented with clustering learners based on their respective number of comments to analyse if grouping as a pre-processing step improved the results. The results show that, whilst multi-input for deep learning can be useful, a better overall effect is achieved by using the transformer model, which has better performance in detecting learners who require intervention. Contrary to our expectations, however, clustering before prediction can have negative consequences on prediction outcomes, especially in the underrepresented class.

Keywords: MOOCs, Intelligent Intervention System, Multi-input Model

1 Introduction

During the COVID pandemic and lockdown, most educational institutions around the world turned to online study [1]. Platforms such as MOOCs became increasingly attractive for a large number of institutions and learners to allow them to continue their studies [2]. Nevertheless, dropout rates on MOOC-based courses can reach 90% [3], which remains, even during the pandemic [4], one of the most long-standing problems of such learning environments [5]. In recent years, many researchers have proposed several solutions to curb dropout rates [6] [7], among them, constructing intervention models to determine learner needs based on identifying urgent comments posted to discussion forums [8] [9]. Interaction with an instructor is considered one of the most important indicators for avoiding dropout in MOOC learners [10]. However, in terms of identifying if intervention is required based on the comments posted in discussion forums as asynchronous communication platforms between learners and instructors [11], due to the huge numbers of comments, instructors cannot effectively monitor, track, identify, and respond to all comments that may require intervention. Therefore, many researchers have attempted to create models to identify comments posted by learners who might need intervention [12] [13]. However, we consider that it might also be helpful to consider the sequence of learners' textual comments (Section 3.2) to reduce dropout rates and improve the quality of the interventions offered.

This study aimed to develop a model to identify learners who require intervention by an instructor based on the sequence of learner comments to predict and mitigate learner dropout on MOOC-based courses. As the absence of interaction and feedback by instructors on discussion forums has been associated with increased dropout rates [10] [14], our objective was to propose an intelligent intervention model. We formalised this challenge as a text classification problem by developing and employing a supervised binary classification model with multiple text inputs based on learner comments. The input consists of the most recent comments of the learner (as further defined in Section 3.2) and the output is the predicted dropout. We applied and trained two recent popular types of classifiers: deep learning [15] and transformer [16], and examined various numbers of inputs for prediction. Therefore, we investigated the following research question (RQ):

RQ. Can we predict learners who may drop out and identify their need for intervention from their most recent comments?

To the best of our knowledge, this is the first study to attempt to identify MOOC learners who may need instructor intervention by using their historical online forum comments as data. The other contribution of this work is that we use a multi-input approach for siamese and dual BERT with binary text classification, with the resulting integrated networks being termed multi-siamese BERT and multiple BERT, respectively.

2 Related Work

The issue of intervention to help learners in MOOC environments is an interesting area for many research communities [17] [18] [19] and an important research direction. In prior literature, instructor intervention has been studied from two perspectives: (i) comments on discussion forums, and (ii) learners.

The use of comments on discussion forums for intervention prediction has received a wide research focus; researchers have attempted to establish different intervention models as a text classification task [8] [12] [20] [13], or used comment features as an input of the classifier [21] [22].

From a learner perspective, prevalent studies have addressed intervention and dropout rates using input characteristics or clickstream data, such as predicting dropouts per week, based on the weekly history of the learner [23]. Also, [19] created a similar weekly prediction mechanism by applying a deep learning approach.

In contrast, there is limited research on intervention based on the comments of learners who are likely to drop out [24]. This is due to the low percentage of learners who enrol on a MOOC course *and* write comments (only around 5–10% [25]). For example, [26] showed that out of 55,013 and 10,190 learners who had registered and enrolled on courses, only 750 and 519 engaged with discussion forums by posting comments, respectively. Among the few pieces of research on this topic, [27] used NLP tools to predict learners who completed a MOOC course with an accuracy of 67.8 %. Other researchers combined clickstream data with discussion forum data. For example, [28] predicted learner completion by employing clickstream data and language in a discussion forum with a 78% accuracy rate.

Furthermore, using sentiment analysis gathered from learners' comments, [29] predicted attrition based on different features including sentiment analysis using a neural network and achieved 72.1% accuracy. Another researcher [30] using the same method predicted dropout rates based on sentiment analysis and clickstream data. Also, [31] found a significant correlation between sentiment and attrition.

As previously stated, this study aimed to develop an intelligent intervention system to reduce learner dropout in MOOC courses. The proposed model is a novel approach that predicts learner dropout (need for intervention) based on learner comments history as a multi-input text classification task to improve instructor intervention and reduce dropout rates.

3 Methodology

3.1 Dataset

The dataset for this research consisted of a real MOOC forum on a Big Data course, Run 2, offered in 2013 and hosted by Warwick University on the FutureLearn platform [32]. This dataset was selected because it contains a high percentage of learners who dropped out due to the difficulty of the topic and some comments did not attract instructor intervention. We collected a total of 5786 comments posted to the discussion forum during the first 5 weeks of the 9-week course, amounting to approximately 50% of the course. Comments from the first half of the course were collected because it is better to intervene at an early stage, before dropout [5]. Exploring the data, it included about 871 active learners, who were defined as those who participate in discussion forums and write at least one text comment (commenters) [33] from 11281 enrolled learners and 4683 accessed learners. Enrolled learners are those who registered while accessed learners are those who both enrolled *and* accessed the course at least once during the first 5 weeks [34]. The number of comments written by active learners varied from 1–209. To create a corpus for all commenters, we collected the history of their comments, as their most recent comments during the first 5 weeks. Then, we defined learners needing intervention as those who dropped out after week 5. To define dropout, we followed the approach of [34] on their weekly prediction of dropout: they supposed that learners are considered to have dropped out if, in the following week, they did not access 80% of the available topics. Therefore, for each learner, dropout was defined as accessing less than 80% of the available topics in week 6, therefore, the dropout rate was 65.9% (574 learners needed instructor intervention) while 34% of learners (297) completed the course.

3.2 Intervention Model

To identify the learners' need for instructor intervention, we propose the general architecture of our prediction model. We implement this model based on the Python library. The input of this model is the most recent sequence of learner comments while the output is the prediction of if a learner needs instructor intervention (dropout) or not.

For the number of inputs, i.e. the most recent sequence of learner comments, we assume that the learner writes multiple comments and that the number of such comments is an unknown value and may differ from one learner to another. Therefore, we need to investigate the optimal (i.e. minimal) number of historical comments that can help to predict dropout. As an initial experiment, we examined an incremental number of comments ranging from 3–7; but, as mentioned, the total number of comments ranged from 1–209.

Then, we clustered commenters into three groups (as we identified that the optimal number of clusters was 3 using the silhouette method), based on the number of comments written, using the Fisher Jenk algorithm [35], as shown below in Table 1. Next, we focused on group 1, as it contains the highest number of learners (797 commenters) and is thus the most representative of the average number of comments a learner writes. Of these learners, 557 (69.8%) dropped out and 240 (30%) completed the course. After that, we repeated the same experiments for the best intervention models using group 1 (797 commenters) with the mean input rounded up from *3.66* to 4 and excluded the other two groups. Please note that group 1 also had the smallest standard deviation (Std).

Table 1. Statistics of each cluster group.

Group	Count	Mean	Std	Minimum	Maximum
1	797	3.66	3.43	1	16
2	65	28.89	12.24	17	62
3	9	108	43.40	71	209

We developed our prediction models based on two main types of algorithms: deep learning and transformer. The reason for using these models is because they represent the cutting-edge in NLP and eliminate the need for specific feature engineering because they can extract features. We will illustrate the two types in the following sub-sections.

Deep Learning. We applied two cutting-edge deep learning algorithms: convolutional neural networks (CNN) [36] and recurrent neural networks (RNN) [37]. For RNN, we used long short-term memory (LSTM) [38], bidirectional LSTM [39], gated recurrent units (GRU) [40] and bidirectional GRU [41]. We split the data randomly into training data and testing data (80% and 20%, respectively, equivalent to 696 and 175 learners, respectively). Then, we split the training data into training data and validation data (80% and 20%, respectively, equivalent to 556 and 140 samples, respectively). Each input was treated as a sub-model before these sub-models were concatenated to build the main model. Lastly, we trained the model using the Adam optimizer (batch size = 64; epochs = 20). The prediction for the output of the final/output layer followed [8] where if the probability value is larger than 0.5, it is deemed positive (1). The outcomes of (0) indicate a potential dropout and urgent intervention is required, and an outcome of (1) represents a completer and no intervention is required.

The general architectures are the same for all models. As a preprocessing step to prepare the data for the input, we built a dictionary for each input that contains unique vocabulary words. To specify the length of the word sequences, we followed [8], constraining the length of each input to 200 words; we also explored and found that most comments were ≤ 200 words, which means just 1.3% of comments were affected by truncation. The shortest sequence was padded by 0 and comments > 200 words were trimmed. The next layer after the input layer is the embedding layer. This layer obtains dense vector representations for words, which we use and fine-tune during training, starting with a pre-trained word embedding 'word2vec' [42] (Word2vec GoogleNews-vectors-negative300). We used 'word2vec' because [8] found that word2vec outperforms GloVe on urgent post (comment) classification tasks. Then the following layers are different, according to the different networks (CNN and RNN).

CNN. The general architecture is shown in Fig. 1. In the convolutional layers, for each input, we applied three Conv1D with 128 units and different kernel sizes (3, 4 and 5) following [8]. These layers go through a rectified linear unit (ReLu) activation, followed by a max-pooling layer to further compress features. Then we concatenate the output from each input. Next, we concatenate all the outputs for all the inputs. This is passed to the dense layer with 64 neurons and ReLu activation. Then, a dropout layer is employed to avoid overfitting [43] as a regularisation technique. Finally, the output layer has 1 unit with a sigmoid activation function because it performs a binary classification task.

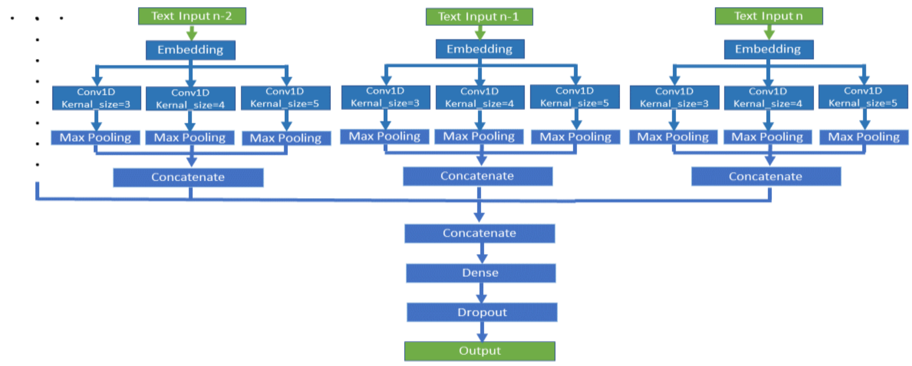

Fig. 1. The general architecture of CNN with multi-input.

RNN. These different networks share the same architecture (see Fig. 2), but the difference between LSTM and GRU is that GRU has fewer parameters. For the bidirectional

LSTM and GRU, we train two layers by adding another hidden layer to reverse to the first layer. Thus, as the next layer after the embedding layer, we have an RNN layer with LSTM or GRU or their bidirectional LSTM and GRU with 128 units. Afterwards, we concatenate the output for each input. Then we add a dense layer and dropout layer as for the CNN. Finally, we move to the output layer with the sigmoid as an activation function to obtain the classification.

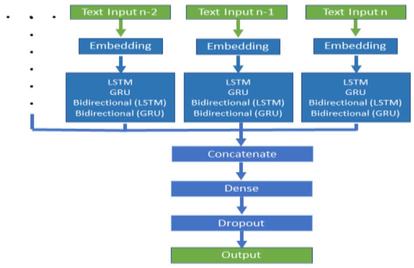

Fig. 2. The general architecture of RNN with multi-input.

Transformer. We developed two different models and built upon the siamese and dual transformers BERT networks to enable the insertion of more than one input into the BERT model [44]. We were inspired to use these two techniques by Marco Cerliani's code on GitHub, which we consequently modified and to which we added more than two inputs (3–7 inputs); additionally, we converted these two multiclass classification models into two binary classification models: multi-siamese BERT and multiple BERT.

The structure of these models is presented in Fig. 3. We convert each text input to transformer inputs and set the maximum length = 202; to compare with the deep learning model, we add two more for special tokens ([CLS] and [SEP]). Then, we utilise BERT base (number of transformer layers = 12, total parameters = 110M), as the training time is less than for the BERT large. We used the same training and testing data as in the deep learning models. Then we train our models using the Adam optimizer, batch size = 6 and epochs = 3. The same for the deep learning model: we calculate the prediction; where if the value is larger than 0.5, it is supposed positive (1). The (0) denotes a potential dropout and needs urgent intervention and (1) denotes a completer and no intervention is required.

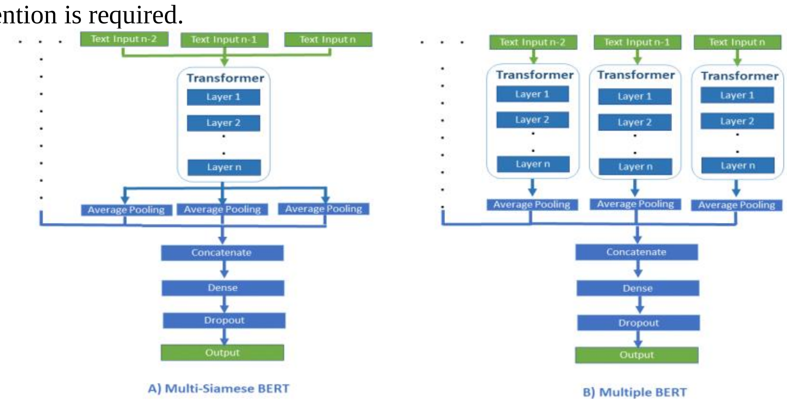

Fig. 3. The general architecture of a) multi-siamese BERT and b) multiple BERT.

Multi-siamese BERT. In this model, the different text input passes to the same transformer. Then, the output is compressed with a global average pooling. After that, we concatenate them and pass them to the dense, dropout, and output layers.

Multiple BERT. In this model, each input passes to different transformers and reduced with average pooling; then we concatenate all the outputs of the global average pooling; after that, as in the multi-siamese BERT, the output is passed to the dense, dropout, and output layers.

4 Results

The experimental results of our multi-input model predictions to address the RQ are presented in Table 2. In addition to accuracy (Acc.), precision (P), recall (R) and F1-score (F1) metrics are also used to comprehensively assess the performance (in percentages) of the different models. In general, the results reveal that LSTM and bi-LSTM achieve high general accuracy but perform badly on the minority class (1), even for the best number of inputs (in italics). GRU also performs badly on the minority class. The remaining models, CNN, Bi-GRU, multi-siamese BERT, and multiple BERT are all more balanced in their optimum models (with the optimal number of inputs). For these best models, multi-siamese BERT and multiple BERT perform comparably and outperform CNN and bi-GRU.

Table 2. The performance of the different multi-input models with different inputs (all learners).

Type		Input	Acc.	0			1		
				P	R	F1	P	R	F1
Deep Learning	CNN	3	0.66	0.76	0.76	0.76	0.44	0.43	0.44
		4	*0.69*	*0.77*	*0.80*	*0.78*	*0.49*	*0.43*	*0.46*
		5	0.67	0.70	0.93	0.80	0.31	0.08	0.12
		6	0.66	0.73	0.81	0.77	0.42	0.32	0.37
		7	0.64	0.74	0.75	0.74	0.40	0.38	0.39
	LSTM	3	0.70	0.70	0.99	**0.82**	0.50	0.02	0.04
		4	*0.70*	*0.70*	***1.00***	***0.82***	***1.00***	*0.02*	*0.04*
		5	0.70	0.70	**1.00**	**0.82**	**1.00**	0.02	0.04
		6	0.70	0.70	**1.00**	**0.82**	**1.00**	0.02	0.04
		7	0.70	0.70	**1.00**	**0.82**	**1.00**	0.02	0.04
	Bi-LSTM	3	0.65	0.74	0.78	0.76	0.41	0.36	0.38
		4	***0.73***	*0.74*	*0.93*	***0.82***	*0.61*	*0.26*	*0.37*
		5	0.67	0.74	0.80	0.77	0.44	0.36	0.40
		6	0.71	0.75	0.88	0.81	0.53	0.32	0.40
		7	0.65	0.72	0.81	0.76	0.39	0.28	0.33
	GRU	3	0.70	0.70	**1.00**	**0.82**	0.00	0.00	0.00
		4	0.70	0.70	**1.00**	**0.82**	0.00	0.00	0.00
		5	*0.70*	*0.70*	***1.00***	***0.82***	***1.00***	*0.02*	*0.04*
		6	0.70	0.70	0.99	**0.82**	0.50	0.02	0.04
		7	0.70	0.70	**1.00**	**0.82**	**1.00**	0.02	0.04
	Bi-GRU	3	0.67	0.74	0.80	0.77	0.44	0.36	0.40
		4	0.63	0.73	0.75	0.74	0.39	0.38	0.38
		5	0.67	0.76	0.77	0.76	0.45	0.43	0.44
		6	0.63	0.76	0.69	0.72	0.42	0.51	0.46

		7	*0.69*	*0.77*	*0.79*	*0.78*	*0.49*	*0.47*	*0.48*
Transformer	Multi-Siamese BERT	*3*	*0.71*	*0.81*	*0.76*	*0.78*	*0.52*	*0.58*	*0.55*
		4	0.63	0.85	0.58	0.69	0.44	0.75	0.56
		5	0.69	0.85	0.68	0.75	0.49	0.72	0.58
		6	0.65	0.85	0.60	0.70	0.45	0.75	0.56
		7	0.65	0.83	0.61	0.71	0.45	0.72	0.55
	Multiple BERT	3	0.67	0.79	0.73	0.76	0.47	0.55	0.50
		4	0.67	0.83	0.66	0.74	0.47	0.68	0.55
		5	0.65	**0.88**	0.58	0.70	0.46	**0.81**	**0.59**
		6	*0.71*	*0.81*	*0.75*	*0.78*	*0.52*	*0.60*	*0.56*
		7	0.67	0.77	0.75	0.76	0.46	0.49	0.48

Thus, we further analysed how the best performing algorithms (transformers) performed for the groups identified, and if our grouping can increase the performance in the given group of focus (group 1).

We can see from Table 3 that the performance of group 1 in multiple BERT outperforms all commenters in multiple BERT in some metrics: (.69%) accuracy, (.92%) recall and (.80%) F1-score for dropout and need intervention (0) but performs badly on the minority class (1) compared to all commenters. Therefore, it provided negative values on prediction outcomes, contrary to our expectations, especially in class (1).

Table 3. The comparison between the performance of different multi-input transformer models with 4 inputs (all learners and group 1).

Type	Group	Acc.	0			1		
			P	R	F1	P	R	F1
Multi-Siamese BERT	All	0.63	**0.85**	0.58	0.69	0.44	**0.75**	**0.56**
	Group 1	0.68	0.70	0.91	0.79	0.59	0.24	0.34
Multiple BERT	All	0.67	0.83	0.66	0.74	0.47	0.68	0.55
	Group 1	**0.69**	0.70	**0.92**	**0.80**	**0.64**	0.25	0.36

5 Conclusion

Although MOOCs offer many learning benefits, they suffer from unacceptable dropout rates. This paper attempted to predict dropout from learners' most recent comments by building ML models including deep learning and transformer with multi-input to enable instructors to intervene more effectively. We developed transformer models based on siamese and dual BERT to insert more than one input for the transformer models. The multi-input consists of the most recent learner comments. We additionally examined the number of inputs needed to predict when intervention was required.

The results indicate that the intervention model can predict dropout and the need for intervention with more accuracy and better detects at-risk learners with the transformer models. However, contrary to our expectations, grouping learners before prediction might harm prediction outcomes, particularly in the minority class. In the future, we plan to replicate this research with other courses and different numbers of comments to further explore the generalisability of these findings. Moreover, we will add clickstream data as additional features.

References

1. Soni, V.D., *Global Impact of E-learning during COVID 19.* Available at SSRN 3630073, 2020.
2. Aljarrah, A.A., M.A.-K. Ababneh, and N. Cavus, *The role of massive open online courses during the COVID-19 era: Challenges and perspective.* New Trends and Issues Proceedings on Humanities and Social Sciences, 2020. **7**(3): p. 142-152.
3. Rivard, R., *Measuring the MOOC dropout rate.* Inside Higher Ed, 2013. **8**: p. 2013.
4. Dang, A., S. Khanra, and M. Kagzi, *Barriers towards the continued usage of massive open online courses: A case study in India.* The International Journal of Management Education, 2022. **20**(1): p. 100562.
5. Cristea, A.I., et al. *Earliest predictor of dropout in MOOCs: a longitudinal study of FutureLearn courses*. 2018. Association for Information Systems.
6. Dalipi, F., A.S. Imran, and Z. Kastrati. *MOOC dropout prediction using machine learning techniques: Review and research challenges*. in *2018 IEEE Global Engineering Education Conference (EDUCON)*. 2018. IEEE.
7. Goopio, J. and C. Cheung, *The MOOC dropout phenomenon and retention strategies.* Journal of Teaching in Travel & Tourism, 2021. **21**(2): p. 177-197.
8. Guo, S.X., et al., *Attention-Based Character-Word Hybrid Neural Networks with semantic and structural information for identifying of urgent posts in MOOC discussion forums.* IEEE Access, 2019. **7**: p. 120522-120532.
9. Alrajhi, L., K. Alharbi, and A.I. Cristea. *A Multidimensional Deep Learner Model of Urgent Instructor Intervention Need in MOOC Forum Posts*. in *International Conference on Intelligent Tutoring Systems*. 2020. Springer.
10. Hone, K.S. and G.R. El Said, *Exploring the factors affecting MOOC retention: A survey study.* Computers & Education, 2016. **98**: p. 157-168.
11. Ramesh, A., et al. *Understanding MOOC discussion forums using seeded LDA*. in *Proceedings of the ninth workshop on innovative use of NLP for building educational applications*. 2014.
12. Sun, X., et al. *Identification of urgent posts in MOOC discussion forums using an improved RCNN*. in *2019 IEEE World Conference on Engineering Education (EDUNINE)*. 2019. IEEE.
13. Khodeir, N.A., *Bi-GRU Urgent Classification for MOOC Discussion Forums Based on BERT.* IEEE Access, 2021. **9**: p. 58243-58255.
14. Wei, X., et al., *A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification.* Information, 2017. **8**(3): p. 92.
15. Young, T., et al., *Recent trends in deep learning based natural language processing.* ieee Computational intelligenCe magazine, 2018. **13**(3): p. 55-75.
16. Vaswani, A., et al. *Attention is all you need*. in *Advances in neural information processing systems*. 2017.
17. Whitehill, J., et al., *Beyond prediction: First steps toward automatic intervention in MOOC student stopout.* Available at SSRN 2611750, 2015.
18. Cobos, R. and J.C. Ruiz-Garcia, *Improving learner engagement in MOOCs using a learning intervention system: A research study in engineering education.* Computer Applications in Engineering Education, 2021. **29**(4): p. 733-749.

19. Xing, W. and D. Du, *Dropout prediction in MOOCs: Using deep learning for personalized intervention.* Journal of Educational Computing Research, 2018: p. 0735633118757015.
20. Almatrafi, O., A. Johri, and H. Rangwala, *Needle in a haystack: Identifying learner posts that require urgent response in MOOC discussion forums.* Computers & Education, 2018. **118**: p. 1-9.
21. Chaturvedi, S., D. Goldwasser, and H. Daumé III. *Predicting instructor's intervention in MOOC forums*. in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014.
22. Chandrasekaran, M.K., et al., *Learning instructor intervention from mooc forums: Early results and issues.* arXiv preprint arXiv:1504.07206, 2015.
23. Kloft, M., et al. *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*. 2014.
24. Prenkaj, B., et al., *A survey of machine learning approaches for student dropout prediction in online courses.* ACM Computing Surveys (CSUR), 2020. **53**(3): p. 1-34.
25. Rose, C. and G. Siemens. *Shared task on prediction of dropout over time in massively open online courses*. in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*. 2014.
26. Gitinabard, N., et al., *Your actions or your associates? Predicting certification and dropout in MOOCs with behavioral and social features.* arXiv preprint arXiv:1809.00052, 2018.
27. Crossley, S., et al., *Language to Completion: Success in an Educational Data Mining Massive Open Online Class.* International Educational Data Mining Society, 2015.
28. Crossley, S., et al. *Combining click-stream data with NLP tools to better understand MOOC completion*. in *Proceedings of the sixth international conference on learning analytics & knowledge*. 2016. ACM.
29. Chaplot, D.S., E. Rhim, and J. Kim. *Predicting Student Attrition in MOOCs using Sentiment Analysis and Neural Networks*. in *AIED Workshops*. 2015.
30. Mrhar, K., O. Douimi, and M. Abik, *A Dropout Predictor System in MOOCs Based on Neural Networks.* Journal of Automation, Mobile Robotics and Intelligent Systems, 2021: p. 72-80.
31. Wen, M., D. Yang, and C. Rose. *Sentiment Analysis in MOOC Discussion Forums: What does it tell us?* in *Educational data mining 2014*. 2014. Citeseer.
32. FutureLearn. Available from: https://www.futurelearn.com.
33. Alrajhi, L., et al. *Urgency Analysis of Learners' Comments: An Automated Intervention Priority Model for MOOC*. in *International Conference on Intelligent Tutoring Systems*. 2021. Springer.
34. Alamri, A., et al. *MOOC next week dropout prediction: weekly assessing time and learning patterns*. in *International Conference on Intelligent Tutoring Systems*. 2021. Springer.
35. North, M.A. *A method for implementing a statistically significant number of data classes in the Jenks algorithm*. in *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*. 2009. IEEE.

36. Kim, Y., *Convolutional neural networks for sentence classification.* arXiv preprint arXiv:1408.5882, 2014.
37. Elman, J.L., *Finding structure in time.* Cognitive science, 1990. **14**(2): p. 179-211.
38. Gers, F.A., J. Schmidhuber, and F. Cummins, *Learning to forget: Continual prediction with LSTM.* Neural computation, 2000. **12**(10): p. 2451-2471.
39. Graves, A. and J. Schmidhuber, *Framewise phoneme classification with bidirectional LSTM and other neural network architectures.* Neural networks, 2005. **18**(5-6): p. 602-610.
40. Cho, K., et al., *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* arXiv preprint arXiv:1406.1078, 2014.
41. Chung, J., et al., *Empirical evaluation of gated recurrent neural networks on sequence modeling.* arXiv preprint arXiv:1412.3555, 2014.
42. Mikolov, T., Q.V. Le, and I. Sutskever, *Exploiting similarities among languages for machine translation.* arXiv preprint arXiv:1309.4168, 2013.
43. Otter, D.W., J.R. Medina, and J.K. Kalita, *A survey of the usages of deep learning for natural language processing.* IEEE Transactions on Neural Networks and Learning Systems, 2020. **32**(2): p. 604-624.
44. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018.