A Reproducibility Study of Deep and Surface Machine Learning Methods for Human-related Trajectory Prediction

Bardh Prenkaj, Paola Velardi
surname@di.uniroma1.it
Sapienza University of Rome

Damiano Distante, Stefano Faralli
name.surname@unitelmasapienza.it
University of Rome Unitelma Sapienza

ABSTRACT

In this paper, we compare several deep and surface state-of-the-art machine learning methods for risk prediction in problems that can be modelled as a trajectory of events separated by irregular time intervals. Trajectories are the abstract representation of many real-life data, such as patient records, student e-tivities, online financial transactions, and many others. Given the continuously increasing number of machine learning methods to predict future high-risk events in these contexts, we aim to provide more insight into reproducibility and applicability of these methods when changing datasets, parameters, and evaluation measures. As an additional contribution, we release to the community the implementations of all compared methods.

CCS CONCEPTS

• **General and reference → Experimentation**; **Performance**; *Metrics*; *Evaluation*; • **Mathematics of computing → Time series analysis**; • **Computing methodologies → Machine learning**; • **Computer systems organization → Neural networks**;

KEYWORDS

trajectory prediction, deep sequential methods, attention networks, educational data mining, health records data mining, reproducibility

ACM Reference Format:
Bardh Prenkaj, Paola Velardi and Damiano Distante, Stefano Faralli. 2020. A Reproducibility Study of Deep and Surface Machine Learning Methods for Human-related Trajectory Prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3412088

1 INTRODUCTION

Trajectory prediction is often associated to moving object analytics [7], however, many other interesting human-related problems can be modelled as a sequence (a "trajectory") of events/actions separated by discrete time intervals. Among these problems are patient health states prediction [18], online student drop-out risk assessment [11, 19, 20], and, more in general, anomaly detection problem [2]. Data in these domains are characterised by: i) an often long sequence of multiple event types, such as diagnosis and medication codes for patients, or exams, tests and other e-tivities for on line students; ii) irregular intervals between recorded events, possibly with unknown delays between the exact time/duration of an event and its recorded time; iii) absence of a precise ordering of events and iv) latent long-term dependencies between events in the sequence. The task consists in either predicting the probability of future events at time $t + 1$, $t + 2 \ldots$, or predicting a single high-risk event, for example, hospitalisation for patients, dropout for students, fraud for online transactions. In the first case, we have a multi-label soft classification problem, in the second, we have a binary label with unbalanced class distribution. Because of its outlined specificities and practical relevance, trajectory prediction is a very challenging task. In the literature, this problem has been addressed leveraging deep sequential methods (see [25] for a survey), more recently enhanced with attention mechanisms [23] to capture long-distance relationships and improve explainability. Deep methods have shown superior performance in many tasks, however, they need a large number of training data to avoid overfitting. Unfortunately, labelled data in many real-world applications - such as those considered in this paper - may be limited and very sparse. Furthermore, due to their highly parametric nature, reproducibility of deep methods is not guaranteed. Reproducibility is concerned with repeating prior experiments based on a specific algorithm in other contexts, for example, in different application domains, with different evaluation measures and characteristics of the datasets. With the continuously increasing amount of contributions in the domain of deep learning, often based on subtle differences and variable combinations of known ideas, reproducibility studies are crucial to provide insight into the actual novelty, reliability, applicability and impact of available methods. In this paper our aim is to analyse several deep methods for trajectory forecasting, on different datasets. For the sake of space, we consider only one predictive task, risk prediction, leaving multi-label classification for future work. Furthermore, we also make publicly available to the community the implementations of all the compared methods, since we verified that only a minority of "deep" methods had available and replicable implementations.

2 EVALUATION METHODOLOGY

2.1 Datasets

As examples of trajectory-shaped risk prediction problems, we consider dropout prediction in online learning platforms, and survival prediction in a critical care tele-health system. We employ two datasets in the first domain, and one in the second. Specifically,

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412088

	Num. of event types (m)	Max. time window length (ℓ) in days	Num. of time series (trajectories)
XuetangX	22	30	23839
KDDCup15	7	25	120542
eICU	9	30	65130

	Avg. per-day sparsity	Avg. day gap between events	Class distribution (0 : 1)
XuetangX	90.02%	10.65 ± 2.07	38.7% : 61.3%
KDDCup15	90.8%	18.06 ± 1.53	20.7% : 79.3%
eICU	73.4%	8.11 ± 7.74	91.45% : 8.55%

Table 1: Datasets characteristics.

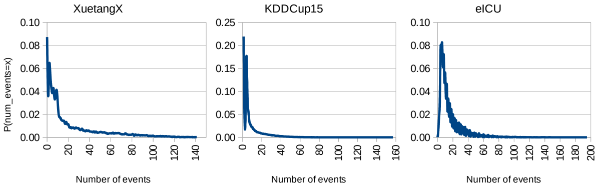

Figure 1: Probability mass function for the number of per-day events for each dataset on max-length sequences

the datasets we use in our study are: *XuetangX*[1], *KDDCup15*[2], and eICU[3]. The first two are benchmark datasets for student dropout prediction and contain e-tivities (homework submissions, video views, navigational actionsb and so on) representing the students' interaction within several online courses. For XuetangX, we prune all e-courses with less than 350 student trajectories, which leaves us with 19 courses overall, whereas we leave KDDCup15 as is, since all courses are sufficiently populated. The medical dataset eICU describes fatalities of patients admitted to multi-centre critical care in US hospitals between 2014 and 2015. Patient-related events include laboratory tests, medications, admissions, physical examinations, and visits. We prune all patient trajectories including zero events, which results in a final set of 65k trajectories. For each dataset, as suggested in [9], we perform a daily grouping of events concerning each individual (student or patient). Therefore, we represent each individual's trajectory u with a time-matrix $\mathcal{T}_u \in \mathbb{R}^{\ell,m}$ where ℓ is the length of the adopted time-window in days and m is the number of different events types (i.e., features) available in the dataset. In detail, a cell $(i, j) \in \mathcal{T}_u$ contains the number of events of type j found in day i of a trajectory u. Table 1 summarises the characteristics of the three datasets in terms of the number of event types m, number of trajectories (time-series), and their maximum length ℓ (see Sect. 2.3 for more details on ℓ). The Table shows that XuetangX has the highest diversity of event types, with a relatively low number of trajectories, whereas eICU has the most imbalanced class distribution, with only 8.55% of fatalities (class 1). We note that, although XuetangX has a mild imbalance, some single courses present acute class disproportion. Furthermore, we have analysed the time-series characteristics for each dataset. In general, all the datasets are composed of a majority of null values in the time-matrices $\mathcal{T}_u$ $\forall u$. An important factor in ordered sequential time-series is the daily gap between events. In Table 1, we also report the average gap between events on the maximum time-window length ℓ and observe that eICU has the highest variance in terms of days. This is a common phenomenon in medical trajectories, since events are not recorded in real-time, as instead happens in the e-learning domains. This is an additional challenge of the eICU dataset since, while in the e-learning domain an absence of events corresponds to sporadic usage of the platform by students and is a relevant predictor of dropout, a gap with absence of events in a medical domain is ambiguous: it may describe a healthy patient (class 0) that needed only some test, or a critical patient that died - class 1 - during his/her hospitalisation. Finally, Figure 1 shows, for each dataset, the probability distribution of finding k daily events, no matter their type, in a randomly selected trajectory u. Although all distributions are zipfian, and KDDCup15 exhibits the steepest curve, the probability of occurrence of at least 1-5 events is much higher (up over 20%) w.r.t. the other datasets. All in all, the data analysis performed suggests that eICU and XuetangX represent more challenging test-beds for trajectory prediction tasks.

2.2 Methods

Based on a literature analysis in the domain of risk prediction on sequential data, we have selected 6 off-the-shelf machine learning models and 6 deep learning techniques[4], grouped as follows:

Simple ML. Here we include *Logistic regression* (LR) with lasso regularisation [21, 24], *Gaussian Naive Bayes* (GNB) without priors [6, 13], *Decision Trees* (DT) with the optimised CART algorithm as in [3, 14, 17], *SVM* with a radial basis function for the kernel [1], and *K-Nearest Neighbours* (KNN) algorithm with two neighbours [13, 27]. Finally, we use a baseline that always predicts the *Majority Class* (MC) of training data.

Ensembles. We use a *Random Forest* (RF) with bootstrapped samples and the Gini impurity to evaluate the splits [8, 9]. Next, Kotsiantis et al. [12] construct an ad-hoc ensemble mechanism - hereafter *KENS* - with majority voting as the consensus function. The base components are WINNOW [16], 1-Nearest Neighbour and Naive Bayes without priors.

Deep feed-forward Neural Network. We implement a three- and five-layered neural network as in [5], respectively DNN-3 and DNN-5. We use a shrinking factor α to calculate the number of neurons: i.e. $n_i = \alpha \times n_{i-1}$ where n_i is the number of neurons in layer i. Inspired by the network architecture in [15], we implement a CNN, namely *Simple CNN*. It consists of two convolution layers (with max pooling) and three dense layers.

Deep sequential. We implement an LSTM [4] - namely *Simple LSTM* - and a combination of CNNs and RNNs - namely *ConRec* [26]. Simple LSTM has m LSTM cells where m is the portion of the sequence length that we consider (see Sect. 2.3 for more detail). We implement *ConRec* by stacking two convolutional layers before passing the output to an RNN. Finally, we feed the hidden vector to a dense layer with a sigmoid activation function.

Deep sequential with attention. First, we incorporate an attention mechanism to a 1d CNN as in *CFIN* [5]. Because XuetangX and KDDCup15 do not contain user-specific data (e.g., grades, exams taken, homeworks submitted) we skip the augmentation and smoothing processes introduced in the paper. Instead, we use an attention context that is initialised according to a uniform distribution. Second, by adapting the *HAN* strategy proposed in [28], we use a local and global attention mechanism in a hierarchical manner with LSTMs.

[1] http://moocdata.cn/data/user-activity

[2] http://data-mining.philippe-fournier-viger.com/the-kddcup-2015-dataset-download-link/

[3] https://eicu-crd.mit.edu/gettingstarted/access/

[4] The implementation of all methods and the benchmark datasets can be found at http://iim.di.uniroma1.it/benchmark.html

2.3 Evaluation Framework

Metrics. Given the imbalanced nature of the data observed in Sect. 2.1, common evaluation metrics are not useful. For instance, accuracy is not informative because a model can always predict the value of the majority class and achieve a high classification score. Additionally, the Receiver Operating Characteristics (ROC) curve is misleading in an imbalanced scenario, as demonstrated in [22]. It follows that the area under the PR curve - hereafter AUCPR - is our chosen metric. Knowing that the majority class differs for XuetangX and KDDCup15 from one course to another, we need to cope with this aspect. Thus, we calculate the precision (P) and recall (R) *weighted by support* (i.e., the number of true instances) for the labels. We, then, use these metrics to construct the PR curve and find the area under it. Lastly, for completeness, we show the average F1 scores to express the goodness of classification for each strategy. Notice that deep strategies output real numbers in the range of [0,1] according to the sigmoid function. Hence, we use thresholds $\theta \in \{0.1, 0.2, ..., 0.9\}$ to binarise the labels and calculate the average R (F1) on all θ. In other words, all the scores that are over θ are set to 1, and those that are under θ are set to 0.

Time-window. Clearly, a real-time detection of the labels is important in the analysed contexts. Hence, we need to use a time-window as minimal as possible. For each dataset we select multiple time-windows of different lengths (i.e. $\ell_1, \ell_2, ..., \ell_n$) and train our models according to them. We expect that the chosen measurements will be monotonically non-decreasing with the increase of the time-window. When training surface machine learning models, we flatten the time-matrices of dimensions (ℓ_i, m), where ℓ_i is the chosen time-window, into a vector of length $\ell_i \times m$.

Training specifics. To analyse the effect on performance of class imbalance - observed especially in the eICU dataset (see Sect. 2.1) - we train our systems both with eICU "as is" and with an oversampling method called ADASYN [10], which is currently reported amongst the most effective. We oversample the minority class using five nearest neighbours for sampling with a balance level of the synthetic samples $\beta = 1$. Thus, we balance the dataset such that the majority and minority classes have the same number of examples. Apart from this, we do not perform any other feature normalisation or pre-processing of any sort before training. The loss function for deep strategies is the binary cross-entropy. We use the ADAM optimiser with a learning rate of 10e-3 with a weight decay of 10e-5 for every deep model except for HAN where we use SGD with learning rate of 0.1 and momentum equal to 0.9. We use 50 epochs for the three- and five-layered networks, LSTM, and the simple CNN. Whereas, we use 10 epochs for ConRec and HAN, 15 epochs for CFIN and 100 for the WINNOW base component of KENS. We specify the batch size for each dataset to be 16 expect for eICU where it is 512. Finally, we follow a 70:30 split for the training and the test sets and average scores on 10 runs. For additional details on parameter settings for all methods, refer to the link where we provide the code and datasets.

3 RESULTS AND DISCUSSION

In order to perform a thorough comparative analysis of all methods, we consider the following aspects:

Performance of deep vs. simple methods with trajectory-shaped data. Because of the latent - though irregular - sequential ordering of events in our datasets, simple machine learning approaches underperform w.r.t. deep sequential methods and deep sequential with attention. Figure 2 illustrates the average AUCPR scores[5] on all time-windows ℓ_i. Notice that, with the increase of ℓ_i, deep strategies perform significantly better, whereas for predictions based on the observation of shorter time-windows they are comparable to simpler models. Moreover, all methods perform better than the *MC* baseline, which indicates that they have successfully learnt to distinguish between the binary classes. In order to assess the general performances and resilience of the classifiers to false negatives, we also show F1 and Recall (Figs. 3 and 4) when varying the threshold θ introduced in Sect. 2.3. Notice the change in trend between the curves[6] in the first two datasets from those in the third. This change in direction happens because the majority class in XuetangX and KDDCup15 is 1, whereas in eICU is 0. Hence, when θ increases, we observe a monotonically non-increasing trend, while in eICU a monotonically non-decreasing trend. The bell-shape curve for eICU with ADASYN indicates that the distribution of posterior probabilities produced by the classifiers follows a uniform distribution with mean approximately equal to 0.5. Therefore, one can think that the number of the posterior probabilities less than 0.5 is equal to those greater than or equal to 0.5. In fact, $P(x < 0.5) = P(x \geq 0.5)$ under a uniform distribution. This explains the fact that the F1 and R scores reach their peak at $\theta = 0.5$ and have mirrored performances for the other thresholds.

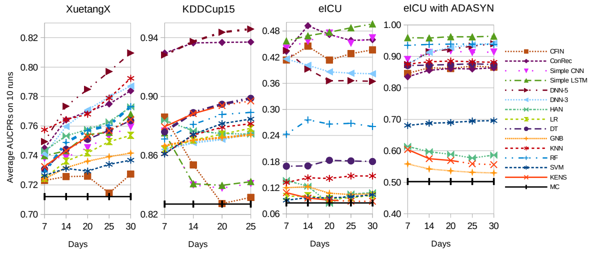

Figure 2: Average AUCPR scores on 10 runs for the datasets.

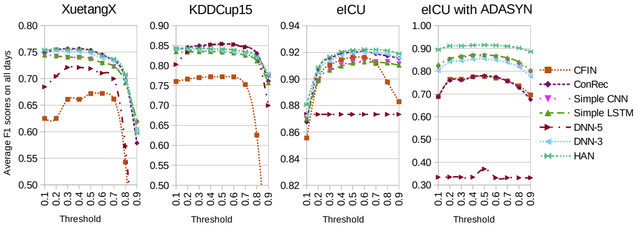

Figure 3: F1 scores averaged on all days varying the threshold θ.

Dataset challenges. As observed in Section 2.1, the trajectories in the medical domain eICU present several challenges, like the variegated number of events per trajectory (Fig. 1), the variance of day-gaps and the high class unbalance (Tab. 1). Because eICU is the most

[5] θ, P and R are calculated as described in *Metrics*, Sect. 2.3

[6] We study only the scores according to the variation of θ for the deep strategies because surface machine learning methods imply a horizontal line as their output (0/1) does not change with θ.

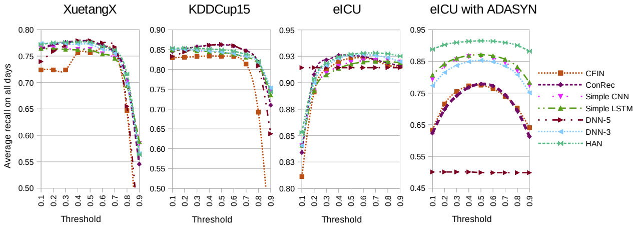

Figure 4: Recall score averaged on all days varying the threshold θ.

challenging dataset, one expects that, for all methods, the average AUCPR scores would be lower than those in the other datasets. In fact, Figure 2 confirms this expectation. Now, consider Figures 3 and 4. Although for the sake of space we do not show precision curves, we can deduce that, since F1 and R are high, but AUCPRs are lower in eICU, then the precision score must be low. On the other side, in risk prediction scenarios, such as that of a critical care system, we can tolerate false positives but not false negatives.

Next, to study the effect of high class imbalance on performance, we augment eICU with ADASYN [10], as detailed in Sect. 2.3. Note in Figure 2 that the class balancing strategy entails a substantial increase (almost 40%) in terms of AUCPRs between the unbalanced and the balanced versions of eICU for all methods.

How the different algorithms cope with dataset challenges. In general, deep feed-forward NNs are the best performing with the exception of the eICU dataset, where Simple LSTM has the best average AUCPR score over all days. The two deep attentive models behave specularly from one another in the two considered domains. In e-learning, CFIN has a decreasing trend, whereas the mean AUCPR of HAN increases when ℓ_i tends to ℓ_n. In the medical domain, we observe a specular behaviour. This happens because of the mediocre sparsity of daily features of eICU (ref. Table 1), which entails that the convolutional component in CFIN has more non-null features to learn from. Whereas, HAN performs poorly because, in the medical domain, events do not have a clear schedule as for student trajectories, that can be naturally divided into weeks or course modules of a specific time-frame. In fact, laboratory and physical examinations and other tests can be done as required and are not always scheduled according to a time-interval. As a consequence, the hierarchical attention strategy does not generalise well since trajectories cannot be effectively divided into sub-trajectories.

Conclusions. We observed that deep methods in general outperform simple machine learning methods and ensembles in problems where data are represented by temporal sequences of events. However, we also demonstrated that the superiority of each deep method over the others is often minimal, while the highest impact on performance is either due to the complexity of input data (in particular, the ratio between the number of different features and the available dimension of training data), or to the application of data engineering methods (e.g., methods to cope with unbalanced classes). Furthermore, different performance indicators may tell a different story: the ranking of systems may significantly change with different evaluation measures, that may be more or less appropriate depending upon the task objectives - e.g., in risk prediction, recall is more relevant than precision - and upon the distribution of data samples. We deduce that the contexts in which one system actually demonstrates its superiority over the others should be investigated more carefully by researchers, using datasets with different characteristics, variable parameters, and different evaluation measures.

REFERENCES

[1] B. Amnueypornsakul, S. Bhat, and P. Chinprutthiwong. 2014. Predicting attrition along the way: The UIUC model. In *EMNLP 2014 MOOCs Workshop*. ACL, 55–59.

[2] R. Chalapathy and S. Chawla. 2019. Deep Learning for Anomaly Detection: A Survey. *CoRR* abs/1901.03407 (2019). arXiv:1901.03407

[3] G. W. Dekker, M. Pechenizkiy, and Jan M. Vleeshouwer. 2009. Predicting Students Drop Out: A Case Study. *Int. Wkng. Grp. on Educational Data Mining* (2009).

[4] Mi Fei and Dit-Yan Yeung. 2015. Temporal models for predicting student dropout in massive open online courses. In *Proc. of Workshop ICDMW 2015*. IEEE, 256–263.

[5] Wenzheng Feng, Jie Tang, and Tracy Xiao Liu. 2019. Understanding Dropouts in MOOCs. In *AAAI 2019*.

[6] E. Gaudioso, M. Montero, and F. Hernandez-Del-Olmo. 2012. Supporting teachers in adaptive educational systems through predictive models: A proof of concept. *Expert Systems with Applications* 39, 1 (2012), 621–625.

[7] H.V. Georgiou, S. Karagiorgou, Y. Kontoulis, N. Pelekis, P. Petrou, D. Scarlatti, and Y. Theodoridis. 2018. Moving Objects Analytics: Survey on Future Location & Trajectory Prediction Methods. *CoRR* abs/1807.04639 (2018). arXiv:1807.04639

[8] C.C. Gray and D. Perkins. 2019. Utilizing early engagement and machine learning to predict student outcomes. *Computers & Education* 131 (2019), 22–32.

[9] L. Haiyang, Z. Wang, P. Benachour, and P. Tubman. 2018. A Time Series Classification Method for Behaviour-Based Dropout Prediction. In *ICALT 2018*. 191–195.

[10] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE IJCNN*. IEEE, 1322–1328.

[11] U. Keshavamurthy and H. S. Guruprasad. 2014. Learning Analytics: A Survey. *Int. Journal of Computer Trends and Technology (IJCTT)* 18(6) (2014).

[12] S. Kotsiantis, K. Patriarcheas, and M. Xenos. 2010. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems* 23, 6 (2010), 529–535.

[13] S. Kotsiantis, C. Pierrakeas, and P. Pintelas. 2004. Predicting Studentsì Performance in Distance Learning using Machine Learning Techniques. *Applied Artificial Intelligence* 18, 5 (2004), 411–426.

[14] Z. J. Kovačić. 2010. Early prediction of student success: Mining student enrollment data. In *Proceedings of Informing Science & IT Education Conference*.

[15] Y. LeCun et al. 2015. LeNet-5, convolutional neural networks. *URL: http://yann.lecun. com/exdb/lenet* 20 (2015), 5.

[16] N. Littlestone and M. K. Warmuth. 1994. The weighted majority algorithm. *Information and computation* 108, 2 (1994), 212–261.

[17] S. Nagrecha, J. Z. Dillon, and N.V. Chawla. 2017. MOOC dropout prediction: lessons learned from making pipelines interpretable. In *Proc. of the 26th Int. Conf. on WWW Companion*. 351–359.

[18] T. Pham, T. Tran, D.Q. Phung, and S. Venkatesh. 2017. Predicting healthcare trajectories from medical records: A deep learning approach. *J. Biomed. Informatics* 69 (2017), 218–229.

[19] Bardh Prenkaj, Paola Velardi, Giovanni Stilo, Damiano Distante, and Stefano Faralli. 2020. A Survey of Machine Learning Approaches for Student Dropout Prediction in Online Courses. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–34.

[20] S. Qu, K. Li, B. Wu, X. Zhang, and K. Zhu. 2019. Predicting Student Performance and Deficiency in Mastering Knowledge Points in MOOCs Using Multi-Task Learning. *Entropy* 21, 12 (2019), 1216.

[21] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H.R Gehlbach. 2016. Forecasting student achievement in MOOCs with natural language processing. In *6th Int. Conf. on LAK*. ACM, 383–387.

[22] T. Saito and M. Rehmsmeier. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* 10, 3 (2015).

[23] H. Song, D. Rajan, J.J. Thiagarajan, and A. Spanias. 2018. Attend and Diagnose: Clinical Time Series Analysis Using Attention Models. In *AAAI-18, IAAI-18, EAAI-18*. 4091–4098.

[24] C. Taylor, K. Veeramachaneni, and U. O'Reilly. 2014. Likely to stop? predicting stopout in massive open online courses. arXiv:1408.3382

[25] S. Wang, J. Cao, and P. S. Yu. 2019. Deep Learning for Spatio-Temporal Data Mining: A Survey. *CoRR* abs/1906.04928 (2019).

[26] W. Wang, H. Yu, and C. Miao. 2017. Deep model for dropout prediction in moocs. In *Int. Conf. on Crowd Science and Engineering*. ACM, 26–32.

[27] W. Xing and D. Du. 2019. Dropout prediction in MOOCs: Using deep learning for personalized intervention. *J. of Educ. Comp. Res.* 57, 3 (2019), 547–570.

[28] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. 2016. Hierarchical attention networks for document classification. In *NACL*. 1480–1489.