Deep learning models and interpretations for multivariate discrete-valued event sequence prediction

Gábor Kőrösi, Richárd Farkas

Institute of Informatics, University of Szeged, Szeged, Hungary

korosig@inf.u-szeged.hu, rfarkas@inf.u-szeged.hu

Abstract. We propose an embedding-based deep learning model architecture for raw clickstream event sequences, which has special characteristics, such as being multivariate discrete-valued. We evaluate the proposed architecture on a Stanford University MOOC dataset, which consists of clickstream-level raw log event data collected during student sessions in the MOOC. We introduce empirical results achieved by various configurations of the architecture on the student final grade regression task. Apart from the regression experiments, we also propose three visual interpretation techniques for explaining the black-box Temporal Convolutional Neural Network and Recurrent Neural Networks models. The goal is to provide easily applicable interpretations which can be used by domain experts without any Machine Learning technical expertise. Based on the visual interpretations, we were able to identify student behavior patterns from raw data, in line with educational research literature.

Keywords: Event log processing, Discrete-valued sequence prediction, TCNN, RNN, interpretations

1 Introduction

User modelling based on users' online behavior has numerous important applications, including recommender systems and educational data mining [1,4,3,12]. In this work, we analyze the online behavior of Massive Online Open Course (MOOC) students. We introduce a deep learning architecture to predict the outcome score of the students at a MOOC. We analyzed a clickstream-level raw dataset which was recorded during a Stanford University MOOC with 142,395 students. A recorded click event of the clickstream-level MOOC usage data consists of four categorical/discrete-valued attributes: action type, description of events, visited link, and the students' success on the subtask in question.

Recently, Neural Networks have been widely used as sequence predictors and time series forecasters, as they can capture complex nonlinear patterns. [9] The most commonly used model is the Recurrent Neural

Network (RNN) which has outperformed statistical models, e.g., auto-regressive and moving-average models. [13] Besides the dominance of RNN models, there have been Convolutional Networks (CNN) proposed for time-series forecasting and sequence classification, namely Temporal CNNs (TCNN). Whereas the majority of the time series deep learning models have been applied to numerical data, event logs, such as click-stream-level MOOC data used in this instance, consists of multivariate discrete-valued sequences. Hence, time series deep learning techniques cannot be directly applied. On the other hand, most of the discrete-valued sequence prediction solutions have been published for Natural Language Processing. The raw event logs are significantly longer than natural language sentences, with their varying length, thus NLP techniques cannot be applied directly. To handle these special characteristics of the given clickstream-level MOOC dataset, we propose an embedding-based deep learning model architecture. In this study, we trained state-of-art RNN and CNN models to predict the outcome score of the students at the MOOC. We conducted experiments using various embedding layers to represent the multivariate discrete-valued data.

Recurrent and Temporal Convolutional Neural Networks provide accurate forecasts without having any access to explicit knowledge about the investigated system. Yet, deep learning methods are typically considered as ‘black boxes’ where it is almost impossible to fully understand what, why, and how RNN and CNN make forecasting decisions. [13] Our research aims to open the black box of RNNs and CNNs trained for time series regression. We propose three visualization techniques, which support domain-expert users in interpreting discrete-valued multivariate time series regression, neural models.

The contributions of this paper are two-fold: 1) we present experimental results on various deep learning architectures and embedding strategies, evaluated on a MOOC clickstream event, discrete-valued time-series regression task, and 2) we propose application-oriented, i.e., user-friendly visualizations for explaining the behavior of the machine-learned RNN and CNN, regression models.

2 Related Work

Analyzing student behaviour in MOOCs directly on the clickstream-level is a new field of study. Li et al. [13] and Baker et al. [3] sought to understand student behavior using log sequence from different MOOC courses. They investigated and visualized behavioral patterns of student groups by employing statistics and classic machine learning methods over hand-crafted features. To the best of our knowledge, the study by Kőrösi and Farkas [11] is the only work to date utilizing deep learning techniques to exploit raw clickstream data which have been recorded during MOOC courses. They reported that they were able to outperform hand-crafted feature-based classic machine learning approaches. In our research, we employ deep learning techniques to solve the same goal as Li et al. [13] and Baker et al. [3], i.e., to analyze student behavior. We can draw educational conclusions similar to those presented in Li et al. [13] and Baker et al. [3], but since we used raw sequences directly, our approach did not require any feature engineering of pedagogical expertise.

Recent advances in neural architectures and their application to raw time-series and sequences offer an end-to-end learning framework that is often more flexible than classic feature engineering-based approaches. [12] For example, Koehn et al. [10] showed that an RNN-based method could outperform common machine learning while using mixed continuous and discrete-valued time series to predict the order value. Guo et al. [6] proposed the feed forward neural network and embedding layer-based DeepFM for multivariate partially raw discrete-valued clickstream data. Apart from the recurrent approaches, convolutional models capable of considering the temporal dimension have recently been proposed. Sadouk [14] proposed an exhaustive study of Convolutional Neural Networks where convolutions were applied in the sequence recognition tasks. Our work was motivated by these studies, thus we experimentally compared CNN and RNN models on discrete-valued sequences.

The embedding of discrete-valued sequences was successfully applied in user behavior analysis[10]. An et al. [2], for instance, presented their neural user embedding approach which was capable of learning informative user embeddings by using the unlabeled browsing-behavior. Cheng et al. [4] introduced the Wide and Deep feature representation method. In our work, we embed our discrete-valued attributes for enhancing the generalization capability of our neural networks.

Karpathy et al. [7] analyzed the interpretability of RNNs for language modeling, demonstrating the existence of interpretable neurons which were able to focus on specific language structures. Siddiqui et al. [15] explored the visualization techniques including input saliency by means of occlusion and derivatives, class mode visualization, and temporal outputs. In Section 5, we applied an approach to interpret our multivariate discrete-valued sequence forecasting model.

3 Dataset

The time-series dataset is made up of raw loglines which have been recorded during the Computer Science 101 online course at Stanford Lagunita University in the summer of 2014. It contains video lectures, optional homework assignments, discussion forums, and quizzes.

Table 1. The Stanford Lagunita's Science 101 dataset

Feature	Examples	No. unique value
Links	'courseware/z187/z172/', 'courseware/z187/z184/'	243
Events	'load_video', 'login', 'problem_check'	34
Resource	'Q1', 'Week 2 Course Survey'	35
Success	0,1,-1* (* missing value)	2

The raw data sequence includes 39.6 million loglines created by 142,395 students. Of these, only 13,574 students completed the course, so we only used the data of these students in our work. On the filtered data each logline is made up of five attributes describing a clickstream level event: event type (categorical variable), visited URL (categorical variable), resource name (categorical variable), and quiz success (binary variable). Table 1 lists some of these examples.

Table 2. Number of logged events in the different progress sections of the course

Event type/progress		20%	40%	60%	80%
Video	Load	34999	67411	97070	127552
	Play	61003	123338	182821	238408
	Seek	18862	41574	61490	80236
	Speed change	3442	5668	7600	9516
Quiz	Quiz 1	20283	42655	73102	110348
	Quiz 2	14581	33294	61091	96684
	Quiz 3	8281	21760	48636	81614
	Quiz 4	46	648	5365	9261

The aim of this research is to predict the student's final scores (from 0 to 100) achieved in the four quizzes based on the raw log sequence. The user could take the quizzes multiple times, but the final score is the sum of the first attempts. To gain a better understanding regarding the users' learning behavior and the predictive power of raw log data, we split the time series into progress sections, namely 20%, 40%, 60%, 80% of the course progress. Table 2 displays the counts of a few event types.

4 Embedding-based Multivariate Sequence Regression

The focus of this study is on multivariate discrete-valued sequence neural regression. We propose a deep learning architecture in our MOOC scenario, which is depicted in Fig. 1-2.

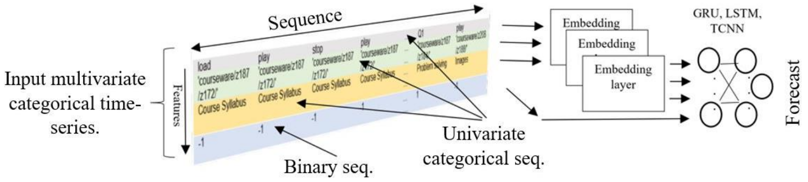

Sequence
Input multivariate categorical time-series.
Features
load 'courseware/z187 /z172' Course Syllabus
play 'courseware/z187 /z172' Course Syllabus
stop 'courseware/z187 /z172' Course Syllabus
-1
Binary seq.
Univariate categorical seq.
Embedding
Embedding
Embedding layer
GRU, LSTM, TCNN
Forecast

Fig. 1. A unified deep learning framework for discrete sequence forecasting. A DL architecture, where the Embedding layers are designed to encode each categorical attribute separately. Then the TCNN and RNN networks learn the hierarchical representations of the sequenced data.

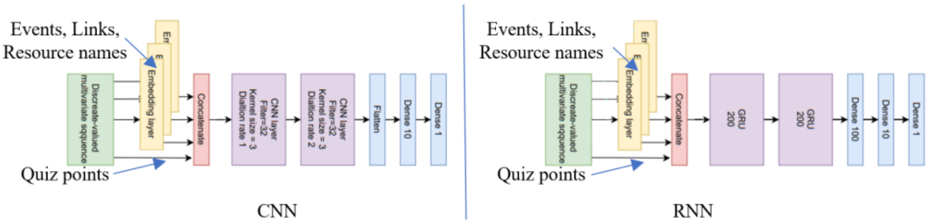

Events, Links, Resource names
Discreate-valued multivariate sequence
Embedding layer
Quiz points
Concatenate
CNN layer Filter=32 Kernel size = 3 Dialtion rate 1
CNN layer Filter=32 Kernel size = 3 Dialtion rate 2
Flatten
Dense 10
Dense 1
CNN
Events, Links, Resource names
Discreate-valued multivariate sequence
Embedding layer
Quiz points
Concatenate
GRU 200
GRU 200
Dense 100
Dense 10
Dense 1
RNN

Fig. 2. Overview of the configurations for multivariate sequence prediction. TCNN architecture is seen on the left, RNN (GRU and LSTM) on the right. The numbers in boxes refers to layer sizes, i.e. number of hidden units.

Embedding layers are designed to encode each categorical attribute separately. Then the TCNN and RNN networks learn the hierarchical representations of the sequenced data. Recurrent and Temporal Convolutional Neural Networks proved their ability to discover patterns in multivariate time-series, giving forecasts without explicit knowledge of the inspected system.[12,25]

Our research aims to create an accurate way to use the same methodology on discrete-valued sequences in RNN and TCNN. Our framework for discrete-valued sequence prediction is depicted in Fig. 2. We propose a representation of discrete sequence in the form of a vector embedding. Instead of any data preparations, we insert the label encoded univariate sequences themself into the embedding layer which could autonomously transform the categorical labels into a continuous space. This has the following advantages: it does not contain any artificial “human-based” parameters which could affect the behavior of the model; while the embedding layer learns without human intervention, it does not strongly depend on how many data points are available; it is suitable for the time-based discrete sequence from high-dimensional attractors.

5 Regression results

We randomly split our student dataset into training- (9502 sequences) and evaluation (4072 sequences) datasets. The mean absolute error (MAE) of final student scores is taken as an evaluation metric.
We calibrated the size of our neural networks on a development set (random subset of the training dataset).

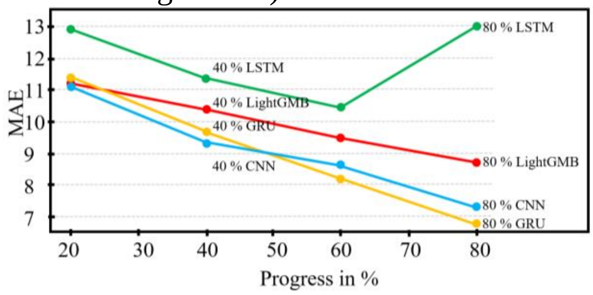


Fig. 3. Mean Absolute Errors achieved by various models at different progress state of the course

We use embedding layers of length 30 (the sizes of other layers are shown in Fig 2). We employ tangent activation function in the GRU and LSTM experiments, while ReLU in the CNN ones. As the optimizer we used Adam with the default 0.0001 learning rate and early stopping criteria. Sequences were post padded to lengths varied in function of student progress datasets (lengths: 20%: 220, 40%: 520, 60%: 720, 80%: 920).
In our baseline model, the user’s behavior in the course is encoded as a 28-dimensional feature vector. These cumulated features consist of the number of video interactions (play, stop, pause), quiz success (quiz 1, 2, 3, 4), etc. We conduct LightGMB regression [8] on the cumulated features as a baseline. Fig. 3 shows that there is no significant difference among the models at 20% progress. The CNN architecture yields either the best, or the second-best performance in most of the data sets.

Table 3. Real (x axis) vs predicted (y axis) final student scores results from LightGMB, CNN, GRU, and LSTM models in different progress point of the course.

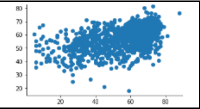
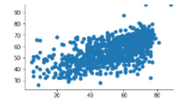
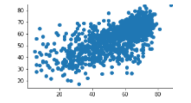
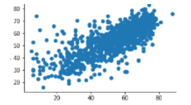
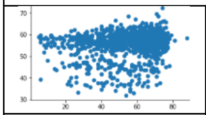
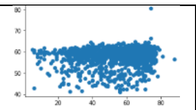
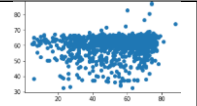
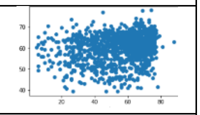
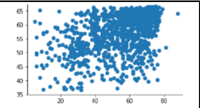
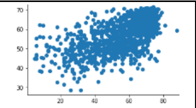
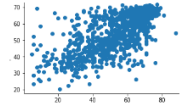
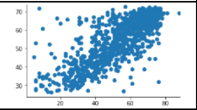
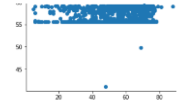
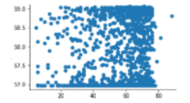
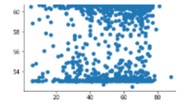
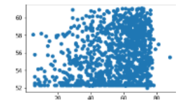
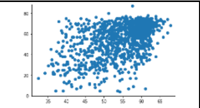
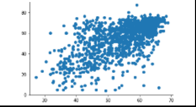
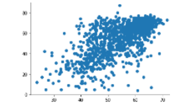
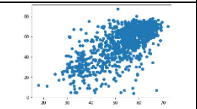
20% progress	40% progress	60% progress	80% progress
CNN - 4 features with embedding			
			
MAE 11.24	MAE 9.47	MAE 8.70	MAE 7.40
CNN - 4 features without embedding			
			
MAE 13.13	MAE 12.84	MAE 12.92	MAE 12.92
GRU - 4 features with embedding			
			
MAE 11.42	MAE 9.81	MAE 8.33	MAE 6.89
GRU – 4 features without embedding			
			
MAE 13.04	MAE 12.96	MAE 13.00	MAE 12.99
LightGMB – 27 (cumulated) feature			
			
MAE 11.34	MAE 10.50	MAE 9.62	MAE 8.82

Table 3 shows that GRU and CNN with embedding has 'captured' the patterns in data better and provided a much better forecast than other implementations. We tested the LSTM with embedding but it generated unmeasurable results. This could be explained by the amount of data because LSTM is sensitive to long sequences, and we have an average of 720 time-steps.

6 Interpretations

Recurrent and Convolutional neural network models have recently obtained state-of-the-art sequence prediction accuracy. However, for data analysis, it remains unclear what the models learned, how these approaches identify patterns and meaningful segments from time-series. This section aims to explore this black box to gain better understanding of the behavior of categorical time series prediction DNN models.

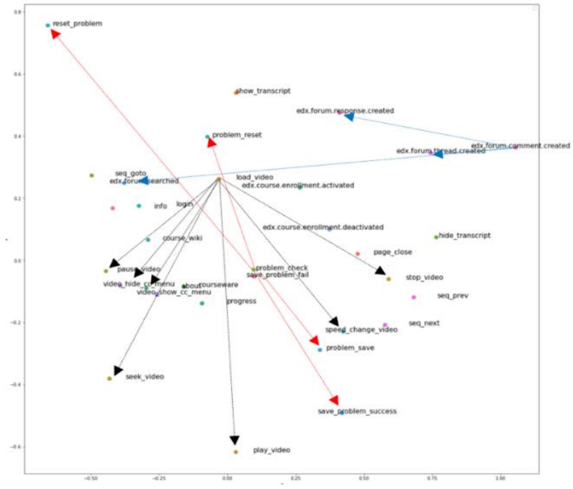


Fig. 4. T-Distributed Stochastic Neighbor Embedding (t-SNE) results for EVENT feature embedding layer. Arrows of different colors represent general groups of different event types.

6.1 Embedding spaces

The MOOC dataset contains four attributes, including three discrete-valued variables. We transfer those three attributes to three parallel embedding layers (See Fig. 1.) to learn and transform discrete-valued values into an nth dimension continuous space.

To understand the trained embedding layers behavior, we used the output of trained embedding layers which was trained on the event attributes, further, we employed t-Distributed Stochastic Neighbor Embedding (t-SNE) so as to map the 30-dimensional embedding space to 2D. Embedding with the t-SNE method is useful because embeddings are learned, thus events, links, or resources that are more similar in the context of our problem are closer to one another in the embedding. The general idea is to group each event type according to its “location” of the curriculum. For example, play, stop, pause would be in the group of video interactions, problem_check, problem_reset, save_problem_succ-es in the quiz group. However, the embedding layer processes this differently. Fig. 4 highlights that both video and forum-based events are coming closer to each other, yet more peculiar is the fact that the save and play video events seem to be similar. The trained embedding layer was able to significantly improve our forecasting results (GRU-based model average increase ~ 8%, CNN-based model average increase ~ 10%), proving to be an effective aid in preprocessing discrete-sequence.

6.2 Temporal saliency

The temporal activity of students during the MOOC is a fascinating pedagogical area to explore. The visualization below indicates how strongly the different temporal segments relate with the deep learning prediction. We also aim to detect whether students with various outcome scores display different temporal behavior.

The RNN and CNN methodology uses the output of embedding layers and one binary attribute to train the models (see Fig. 2). As a result of the training process, we use the output of CNN and RNN layers with the absolute value of the derivative of the loss function with respect to each dimension of all sequence inputs. Each row in Fig. 5. corresponds to the predicted student outcome group. Since very few users made up the first group (0-10 final student scores) and the last group (80-90 final student scores), this interpretation was omitted. The columns in the figure represent the output of CNN and GRU layers as the mean of the loss values. By visual inspection of the mean of the loss function values, we can see from the heat map (Fig 5.) that CNNs tend to focus on short contiguous subsequences ("windows/boxes") when predicting the outcomes, whereas GRU uses the whole sequence for the same task. In other words, CNN's model finds "motifs" that are important for prediction, by comparison, GRU apparently gives a different gradient for each time step. The results are almost the same as seen in Lanchantin et. Al. (2017), in their research about using CNN and RNN to understand DNA sequence. They found that the recurrent neural network tends to be spread out more across the entire sequence, indicating that they focus on all sequences together, and infer relationships among them. They also mentioned that, when using convolutional and recurrent networks for sequence forecast, those tended to have strong heat points around motifs, where one could see that there were other steps further away from the motifs that were significant for the model results. Both CNN and GRU have a considerably wide range of steps, moreover, for the low outcome final student scores (0-40) the RNN model uses the entire sequence, while for high final student scores it uses only the first part of the dataset. CNN uses windows of almost the same size as all outcome classes, and although the distribution of weights is different, it learns from the middle of sequences which is completely different from RNN.

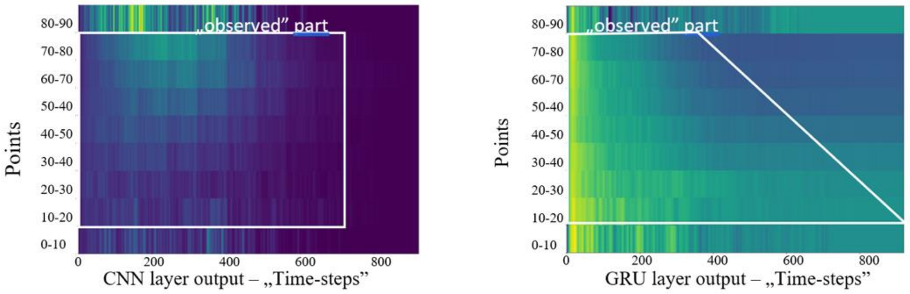


Fig. 5. Representations over time from CNNs and GRUs layers. Each row corresponds to the predicted student result group from CNN and GRU at each time-step. Each grid from the column corresponds to each dimension of the current sequence step representation. We observed only that part of the heatmap, where the data is not constant, or not too uniformly distributed. The brighter color means high activation at the output of the layer of our neural network, even the dark means weak activation.

6.3 User behavior clustering

In order to identify the different learning strategies and examine whether they appear in the data sequence, we conducted further studies. We investigated the best and worst 20% of student groups. We conducted a cluster analysis (Kmeans, n_clusters = 2, algorithm=Elkan), utilizing the hidden vector representation learned by our CNN and GRU models. The clustering is based on the cosine similarity of the output 50-dimensional vectors of the CNN and the GRU layers. As an interpretation of the clusters, the features introduced in Section 3 were accumulated from the cluster members. Fig. 6 shows the boxplots of the key features by clusters. The results of the best and worst 20% clustering show that there are two different clusters among both top and worst-performing students. The first group (marked in blue) watch significantly fewer videos (rdn/Video) than the others do while achieving the same result. The feature values describing the interaction between the users and videos (numoplay_video, numostop_video, numopause_video, numoseek_vi deo) also underpin this observation. Our click-stream level raw data-driven results are in line with educational/pedagogical results. For example, Galine et. al. [5] sought to understand the behavior patterns of learners in MOOC courses and they found that at the very base level, there were "All-rounders" and "Viewers", the terminology being similar to the results of our unsupervised clustering analysis: users marked blue seem to be “All-rounders”.

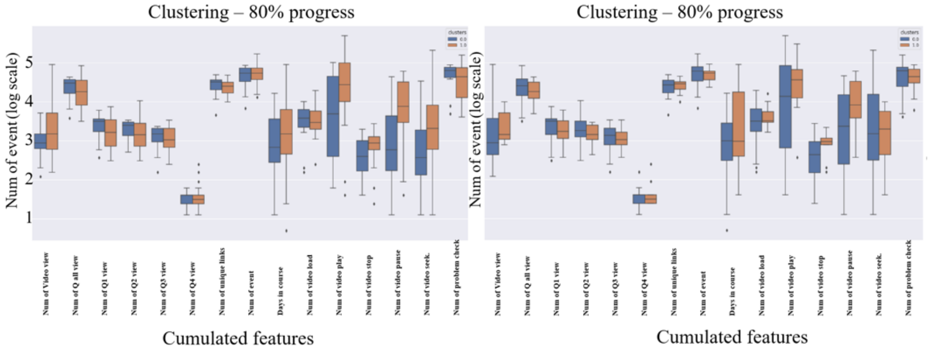


Fig. 6. Cluster analysis of the group of 20% -20% students who achieved best (left) and worst (right) final student scores during the course. The blue and orange colors are show the different clusters in the observed group.

The blue cluster members complete most assignments, watch all video lectures, and have numerous interactions with video, while “Viewers” (brown cluster) watch almost all video lectures but hardly ever make more effort than absolutely necessary to complete the course. This data-driven interpretation of MOOC log data is a promising direction for educational data mining, as we were able to show sociological-pedagogical results using only raw logline data, which has not been seen before.

7 Conclusion

Our literature review established that the existing deep learning-based time-series prediction models could handle both continuous and discrete-valued sequences. In this work we proposed RNN and CNN based methods with embedding-based deep learning model architecture which is able to make a prediction from multivariate discrete-valued, variable-length sequences. The models were tested on a Stanford University MOOC dataset, which consisted of clickstream-level raw log event data collected during student sessions in the MOOC. Our results confirmed that RNNs and CNNs provided a better forecast than conventional methods. The interpretation section outlined that the embedding method was able to significantly improve our forecasting results and provide an effective aid in unsupervised pre-processing of discrete-valued sequence.Besides creating accurate methods, we also proposed three useful visualization of the learnt deep neural networks.

Acknowledgement

This work was supported by the National Research, Development and Innovation Office of Hungary through the Artificial Intelligence National Excellence Program (grant no.: 2018-1.2.1-NKP-2018-00008) and the framework of the Artificial Intelligence National Laboratory Program.

References

1. Aldowah, H., Al-Samarraie, H..: Educational data mining and learning analytics for 21st century higher educationIn: Telematics and Informatics, 37, pp. 13–49 (2019)
2. An, M., Kim, S: Neural User Embedding from Browsing Events. In: Dong Y., Mladenić D., Saunders C. (eds) Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track. ECML PKDD 2020 (2020)
3. Baker, R., Xu, D., Park, J. et al.: The benefits and caveats of using clickstream data to understand student self-regulatory behaviors. In: Int J Educ Technol High Educ 17 (2020)
4. Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, T., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah H.: Wide & Deep Learning for Recommender Systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (2016)
5. Galina, M., Daria, M., Kristina, Z.: Correlation of MOOC Students' Behavior Patterns and Their Satisfaction with the Quality of the Course. In: Proceedings of the "New Silk Road: Business Cooperation and Prospective of Economic Development", pp 5282-5291 (2019)
6. Guo, H., Tang, R., Ye, Z., Li, Z., He. X.: DeepFM: a factorization-machine based neural network for CTR prediction. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1725–1731 (2017)
7. Karpathy, A., Johnson , J., Fei-Fei Li, F.: Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078 (2015)
8. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma,W., Ye, Q., Liu, T.: LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17) pp. 3149–3157 (2017)
9. Kedem, B., Fokianos, K.: Regression Models for Time Series Analysis (2002)
10. Koehn, D., Lessmann, S., Schaal, M.: Predicting online shopping behaviour from clickstream data using deep learning. In: Expert Systems with Applications (2020)
11. Kőrösi, G., Farkas R.: MOOC performance prediction by Deep Learning from raw clickstream data. In: Advances in Computing and Data Sciences, pp.474–485 (2020)
12. Lee J.M., Hauskrecht M.: Recent Context-Aware LSTM for Clinical Event Time-Series Prediction. In: Artificial Intelligence in Medicine (2019)
13. Li, Q., Baker, R., Warschauer, M.: Using clickstream data to understand, and support self-regulated learning in online courses. In: The Internet and Higher Education (2020)
14. Sadouk, L., Gadi, T., Essoufi, E.H., Alonso-Betanzos, A.: A Novel Deep Learning Approach for Recognizing Stereotypical Motor Movements within and across Subjects on the Autism Spectrum Disorder. Intell. In: Neuroscience (2018)
15. Siddiqui, Shoaib & Mercier, Dominique & Munir, Mohsin & Dengel, Andreas & Ahmed, Sheraz. (2019). TSViz: Demystification of Deep Learning Models for Time-Series Analysis. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2912823.