Early prediction of students' efficiency during online assessments using a Long-Short Term Memory architecture

Cristina Villa-Torrano, Miguel L. Bote-Lorenzo, Juan I. Asensio-Pérez,
Eduardo Gómez-Sánchez

GSIC-EMIC Research Group, Universidad de Valladolid, Valladolid, Spain
cristina@gsic.uva.es
{migbot,juaase,edugom}@tel.uva.es

Abstract. Nation's Report Card Data Mining Competition 2019 (NAEP Competition) aims to understand which students' behaviors are effective or ineffective during online assessments and to determine how quickly these behaviors can be detected. Specifically, the ultimate purpose is to be able to classify students as effective or ineffective based on the logs of their actions on the National Assessment of Educational Progress (NAEP), the largest nationally assessment of what America's students know in various subject areas. To solve this challenge, our proposal is based on modeling the evolution of student behavior throughout the assessment, considering different characteristics such as the sequence of activities performed and the order in which they have been carried out. The proposed classification model is based on the Long-Short Term Memory (LSTM) recurrent neural network architecture, as it is capable of capturing evolutionary patterns over time. This architecture has been evaluated with the competition dataset and the results obtained are shown, which are very promising.

Keywords: Clickstream - Deep Learning - Long-Short Term Memory - Online Assessments - NAEP Competition

1 Introduction

Due to the great development of online educational platforms, such as Massive Open Online Courses (MOOCs) or mobile applications, large amounts of educational data of a very diverse nature are currently being generated: log sequences, audios, videos, ... [13]. The data generated through online platforms describe the actions of the students in the context in which they occurred with a granularity of seconds between actions ("micro-level data"). The nature and the granularity of micro-level data makes it ideal for real-time interventions, as it is often used to detect cognitive strategies, affective states or self-regulated learning behaviours [4]. Therefore, the treatment and understanding of these data is very useful to improve learning.

As a consequence, there is a huge increase in the development of tools based on Learning Analytics (LA) [7] and Educational Data Mining (EDM) [3]. These tools can be used to solve different problems, such as predicting dropouts [12], or detecting different students' behaviours to support them with personalised recommendations [11].

One specific problem that is attracting the attention of the research community is the detection of students' behaviour during online assessments through eye-tracking technology, response time procedures, etc [5]. In this work, we make early predictions about student efficiency while doing online assessments. We want to detect if students are gaming the system or if they are carrying out misleading strategies. The context is determined by our participation in the NAEP Competition [2], where more than 80 individual and teams from all over the world have participated.

Concerning our solution, we conducted an analysis on the competition's dataset and found important characteristics that potentially classified the students as effective or ineffective at performing assessments. We observed that the sequence of activities performed, as well as the order in which they were done, were good predictors. Accordingly, we suggested to use the Long-Short Term Memory (LSTM) model, as it is capable of capturing evolutionary patterns over time. In recent years, the use of such neural networks in the field of EDM has increased. For example, in [10] they use an LSTM architecture to enable real-time adaptation in MOOCs by recommending the next resource to visit in a personalised way; while in [8] an architecture based on LSTM is proposed to predict student performance.

The rest of the article is divided as follows. The purpose of the competition and the available dataset are described in Section 2. The Section 3 explains data transformations and feature selection. Then, Section 4 present the architecture of the model used in the competition and its implementation. Finally, in Section 5 the results are shown and some conclusions are outlined.

2 NAEP Data Mining Competition 2019

The NAEP Competition [2] aimed to understand which behaviors are effective or ineffective in performing assessments and to determine how quickly these behaviors can be detected. To this end, the proposed dataset is part of the American national test known as *National Assessment of Educational Progress (NAEP)*. This dataset is a compilation of the student actions taken during the mathematics test in the 2016/17 academic year. Specifically, students worked on two blocks of math problems, referred to as Blocks A and B. Each block contained a certain number of problems and the students had a maximum time of 30 minutes to complete the problems in each block. Once those 30 minutes were up, the students could not perform any more actions in that block.

Accordingly, the final purpose of the competition was to make a classifier to determine if the students would act efficiently in Block B by having only the sequence of actions performed in Block A.

Furthermore, for this competition, efficient behaviour is defined as follows:

1. Be able to complete all the problems in block B.
2. Be able to allocate a reasonable amount of time (they said: the minimum possible) to solve each problem, using the 5th percentile as the cut-off.

2.1 Dataset description

The dataset was divided into 6 different files, which include the following ones:

- data_a_train.csv: Contains the logs of the actions performed by each student in Block A. It is part of the training dataset, with 1232 students and a total of 438.291 interactions.
- data_a_hidden_10.csv: Contains the actions performed by the students in their first 10 minutes of activity in Block A. It is part of the test dataset, with 411 students and a total of 47.563 interactions.
- data_a_hidden_20.csv: Contains the actions performed by the students in their first 20 minutes of activity in Block A. It is part of the test dataset, with 411 students and a total of 110.481 interactions.
- data_a_hidden_30.csv: Contains the actions performed by the students during the first 30 minutes of activity in Block A. It is part of the test dataset, with 410 students and a total of 143.880 interactions.
- data_train_label.csv: Contains the target variable of students in the training set.
- hidden_label.csv: Contains the order in which the predictions must be submitted.

The information provided in the first four files is presented in Table 1. There are 7 different attributes, which are described.

3 Data transformation

3.1 Preprocessing

As we mentioned previously, the raw dataset contained sequences of actions performed by each student labeled with the timestamp of the moment in which they were performed. In order to extract the desired characteristics to build the classifier, the following transformations were first performed:

- Rows without a timestamp were removed
- The problems/items were coded as integers
- The 25th, 50th and 75th percentiles[1] were calculated, as well as the upper and lower outliers, for the following characteristics:
  - Time spent by each student for each activity
  - Time spent for each type of activity

[1] Percentiles and outliers were calculated using the training and tests sets

Table 1. Attributes provided in each of the datasets

Attributes	**Description**
STUDENTID	Unique identifier for each student
Block	It is the block to which the action belongs. In this case, we only have the actions of block A.
AccessionNumber	Unique identifier for each of the problems/items. There are 24 different problems/items.
ItemType	The type of problem/item, e.g: multiple-choice question. There are 10 different types of problems/items.
Observable	The type of action the student performed, such as: Click option, delete option, open calculator, etc. There are 24 different types of actions.
ExtendedInfo	Additional information on the action performed, such as the option the student clicked. There are 23.725 different records.
EventTime	The timestamp at which the action was performed.

- Use of the support functions. According to the study carried out in [5], the support functions can be observed in the “Observable” attribute. Cognitive processes associated with the use of these functions can be extracted from the records. An example of this could be the number of times a student “opens the calculator” on the platform and the time she uses it per exercise.

3.2 Feature selection

Considering the nature of the problem, we considered that it would be important to capture the evolution of time per activity per student, as well as the order in which they were performed, and the use of the support functions over time. Therefore, once the preprocessing was done, we carried out transformations to produce the features shown in Table 2.

4 Long-Short Term Memory

Since we were interested in capturing the evolution of students’ behaviour over time, the model selected was the *Long Short-Term Memory* (LSTM) [6]. The LSTM is an extension of the classical *Recurrent Neural Networks* (RNN) [9] where a hidden state and a ‘long-term cellular state’ are maintained. These extensions have made it a good classifier when having patterns determined by very long sequences.

Specifically, the architecture we used to carry out the classifier is based on those proposed by [8] and [10]. Figure 1 shows an scheme. In this scheme it is possible to see how the sequences of actions are introduced to the architecture and transferred to an Embedding layer. The Embedding layer is intended to map discrete (categorical) variables to continuous number vectors. It is used to

Table 2. Features produced for the model

Features	Description
Input_1	Sequences of integers representing the activities performed by each student in order of completion. The order of completion is determined by the timestamp of the last action performend in each activity. For example, if the actions were represented by A_x where x determines the number of the activity and we had the following sequence: A_1, A_1, A_2, A_2, A_1 the sequence of activities would be [2,1], since the last action performed was on activity 1.
Input_2	Sequence of integers representing the percentile for the time spent on each activity in the order indicated above.
Input_3	Sequence of integers that represent the percentile for the time spent on each of the support function.
Input_4	Sequence of integers that represent the percentile of the time used by type of activity.

reduce the dimensionality of the variables and learn a meaningful representation of the categories in the transformed space. In the case of this competition, we were looking for the Embedding layer to find a representation of the different behaviours along the sequences. Therefore, the LSTM layers would be able to detect patterns over time, taking into account the order of the actions being performed. Thereafter, the layer that remains to be highlighted is the Global-MaxPooling (GMP) layer, which aims to reduce the size of the space in which the different variables are represented. Hence, the number of necessary parameters is reduced. Furthermore, as detailed in [8], it increases the predictive capacity, especially in unbalanced datasets.

4.1 Implementation

Evaluation measures To evaluate the predictions made by the classifiers, the competition organizers used two metrics: the area under the curve (AUC) and Cohen's Kappa, both of which were adjusted. The AUC is a robust metric for evaluating a binary classifier, since it considers the relationship between the false positive and false negative rate according to a discrimination threshold. The aim is to maximize the value of this metric. On the other hand, Cohen's Kappa is a statistical measure that adjusts the effect of hazard on the classification made.

The adjustment made for each of the metrics was the following:

$$AdjustedAUC = \begin{cases} 0 & if\,AUC < 0.5 \\ 2(AUC - 0.5) & otherwise \end{cases}$$

$$AdjustedKappa = \begin{cases} 0 & if\,kappa < 0 \\ kappa & otherwise \end{cases}$$

The final result of the evaluation was calculated using the aggregate score of both metrics:

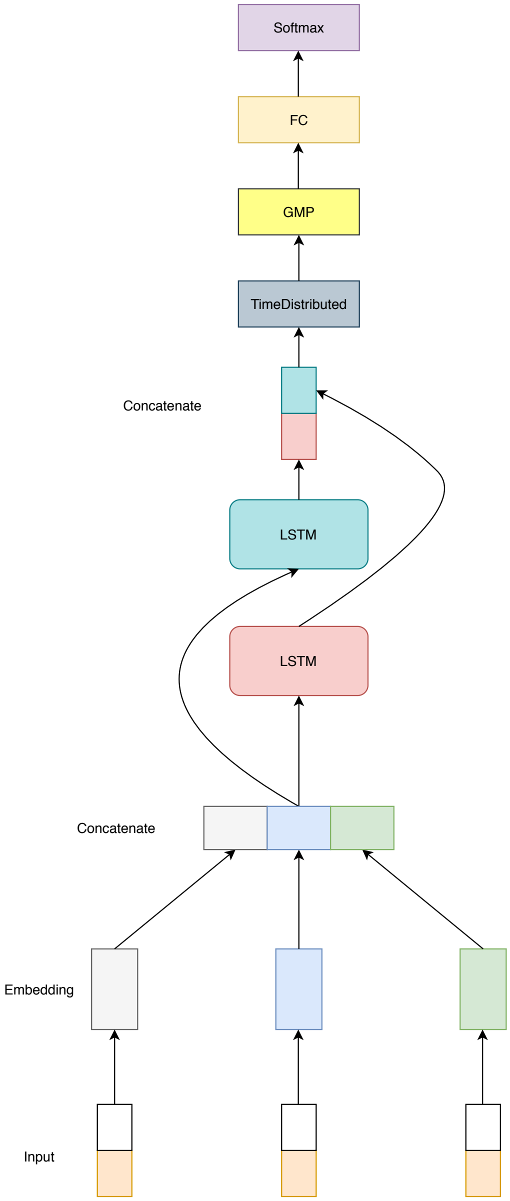

Fig. 1. Deep learning architecture for classifying efficient students during online assessments

$$Final\_Score = AdjustedAUC + AdjustedKappa$$

Training The classifier was trained following a 5-fold cross-validation process at student-level. Since we worked with a binary classifier, the selected loss function, which we had to minimize during the training, was the *binary cross entropy* using the *Adadelta* optimizer and the metrics mentioned above.

In our experiments, we used the BLSTM with forward and backward LSTM layers with a total of 64 units per layer. The dropout rate was set to 20% and applied to both the output of the BLSTM and the output of the TimeDistributed layer. These values were determined by doing a hyperparameter search. Following the recommendation of [1], the Embedding layer had the number of possible values per characteristic as input and the 4th root of this maximum value as output.

Finally, as we had to classify the behaviour of the students having limited actions of them (first 10 minutes, first 20 minutes and 30 minutes), we chose to train three different models, one for each group.

5 Results and discussion

The training results without adjustment are shown in Table 3. As expected, the results are improving as we have more data available from the students. The results can be improved, especially for the model of the first 10 minutes of the assessment, but we still obtained competitive results, as we were sixth in the competition. The main problem we faced was underfitting. One of the possible reasons of this underfitting may be that we generated few features per student that summarized their behavior. LSTMs are often used with raw data, allowing the architecture to discover the features and the relationships between them [10]. Therefore, in future work we would like to explore and exploit the potential of the LSTMs with raw data.

Table 3. Training results without adjustment

	AUC	**Cohen's Kappa**
10 minutes	0.6281 ±0.0290	0.1815 ±0.0545
20 minutes	0.6801 ±0.3150	0.2430 ±0.0353
30 minutes	0.7130 ±0.0371	0.3411 ±0.0833

6 Acknowledgments

This research is partially funded by the European Regional Development Fund and the National Research Agency of the Spanish Ministry of Science, Innovations and Universities under project grants TIN2017-85179-C3-2-R.

References

1. Introducing tensorflow feature columns. https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html, [last access: May 2020]
2. Naep data mining competition. https://sites.google.com/view/dataminingcompetition2019/home?authuser=0, [last access: May 2020]
3. Baker, R.: Challenges for the future of educational data mining: The baker learning analytics prizes. Keynote talk at the 9th International Conference on Learning Analytics and Kowledge 11(1) (2019)
4. Fischer, C., Pardos, Z., Baker, R., Williams, J., Smyth, P., Yu, R., Slater, S., Baker, R., Warschauer, M.: Mining big data in education: Affordances and challenges. Review of Research in Education 44 (03 2020)
5. Hicks, J., Circi, R., Li, M.E.: Students' use of support functions in dbas: Analysis of naep grade 8 mathematics process data. In: Proceedings of the 12th International Conference on Educational Data Mining. pp. 568–571 (2019)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
7. Joksimovic, S., Kovanovic, V., Dawson, S.: The journey of learning analytics. HERDSA Review of Higher Education 6, 37–63 (2019)
8. Kim, B.H., Vizitei, E., Ganapathi, V.: Gritnet: Student performance prediction with deep learning. https://arxiv.org/abs/1804.07405, [last access: May 2020] (2018)
9. Medker, L., Jain, L.: Recurrent neural networks. Design and Applications (5) (2001)
10. Pardos, Z., Tang, S., David, D., Le, C.: Enabling real-time adaptivity in moocs with a personalized next-step recommendation framework. In: Proceedings of the 4th ACM Conference on Learning @ Scale. pp. 23–32. ACM (2017)
11. Schiaffino, S., Garcia, P., Amandi, A.: eteacher: Providing personalized assistance to e-learning students. Computers Education 51(4), 1744–1754 (2008)
12. Wang, W., Yu, H., Miao, C.: Deep model for dropout prediction in moocs. In: Proceedings of the 2th International Conference on Crowd Science and Engineering. pp. 26–32 (2017)
13. Yeung, C.K., Yeung, D.Y.: Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction. International Journal of Artificial Intelligence in Education 29 (05 2019)