

# Student Testing Activity Dataset from Data Structures Course

Paul Stefan Popescu, Marian Cristian Mihaescu, Oana Maria Teodorescu, Mihai Mocanu

University of Craiova

Department of Computers and Information Technology

{spopescu, mihaescu}@software.ucv.ro, teodorescuoanamaria@yahoo.com, mocanu@software.ucv.ro

DOI: 10.37789/rochi.2020.1.1.24

## ABSTRACT

E-learning platforms became more and more popular not only for distance learning but also for learning in full-time education. As this popularity grows, we can use the data extracted from them to complement the professor's work and make predictions regarding students' performance. In this paper, we present a dataset extracted from our e-Learning platform, which is based on the logs collected from testing activity. The focus of this paper is to present the dataset; the experiments presented in the paper are meant to explore the dataset along with its capabilities. The dataset consists of attributes relevant to the testing activity and provides labels which consist of average test grade and final exam grade. Our focus when building the dataset was to keep only the attributes relevant for the learning activity and to provide means to analyse and predict the student's final grade or failure. The paper presents the structure of the dataset, the methodology of collecting the data and experiments using several popular algorithms. The experimental results reveal that the actions performed by the users correlate with the results of the tests and the exam failure can be predicted with a pretty good accuracy using the default set of tuning parameters for our algorithms. As feature work, we can extend the set of experiments with other algorithms, and we can also use parameter tuning for each algorithm for a slight increase in performance.

## Author Keywords

Machine Learning, Educational Dataset, e-Learning, User Modelling

## ACM Classification Keywords

H.5.m. Information interfaces and presentation: Miscellaneous.

## INTRODUCTION

This paper refers to the area of educational datasets which can be used for machine learning tasks. Even if the research area of educational data mining started a long time ago, there are few public educational datasets which can be used as a reference when classifying or predicting the student's performance. The main problem when referring to the prediction of student's performance is a lack of a generic and reproducible approach which may lead to a relevant and generic result. We aim to produce a dataset which can be

continuously updated and can help instructors to early predict student's failure using machine learning algorithms.

In this paper, we present a dataset extracted from the testing activity logged in our e-Learning platform which offers all the required functionalities for online education. The features extracted for this dataset are referring the testing activity and they can be gathered from most e-learning platforms available these days. After presenting each feature of the dataset along with several metrics which can provide a deeper understanding of the data, we conduct some experiments using some common machine learning algorithms. Even though we use an e-Learning platform for taking tests, students who contributed with their actions are from full-time education programs and taking tests was part of the requirements of the Data Structures course in which they were enrolled.

The e-learning platform was custom implemented for running at our University mainly for research purposes and has four roles implemented: student, professor, secretary and administrators. The main functionalities of the platform are learning resources management, communication (between students, students and professors and students and secretaries), testing and live presentations. The learning resources management module offers courses management, homework and external references; the course management option includes the testing setup, which makes the subject of this paper. The platform was designed to be easy to use and to log most of the user actions in order to provide useful insight regarding the activity performed and also to provide relevant research data.

Despite the differences between our platform and other online educational environments, the main features of the platform are quite generic and the structure of the dataset can be obtained from most of the other platforms, so the results are relevant for a large number of researchers. The novelty that comes with this dataset is that we logged all the information regarding the testing procedure, so the grade which is also the class (or the value we want to predict) benefits from several relevant attributes. Another aspect regarding the class attribute is that we computed the mean grade obtained from the tests, which can be one of the classes. However, we also added the grade obtained at the final exam so we can analyse and compute how much taking tests influences the final grade.

The primary motivation of computing and publishing such a dataset is that we can predict the student's failure before the exam will take place even if this is not an evolutionary approach. The reason is that in our scenario, the students took tests before the semester ended and they had to take the exam after that so that the test's grade may be a relevant indicator for their final result. There is also a correlation between the tests they took and the final exam as the questions used in the tests referred to the same topics they have to learn for the final exam. Even though the topics were the same, there are several differences between the tests they took and the final exam as the tests have questions from only a part of the topics necessary at the exam.

### RELATED WORK

One of the most used and referred datasets in educational data mining literature is [1] which logs 30 attributes about students and offers three grades: G1, G2 and which corresponds to the grades obtained during first and second periods (or semesters) and G3 which is the final grade. The primary dataset which is stored at UCI Machine Learning Repository consists of two csv files which can be used together or separately; one of it logged the data for a mathematics course, and the other one logged the data from a Portuguese course. The main catch of the dataset is that it offers many attributes, mostly demographic about the students and final grades, without giving a better insight into how well a student performed in the educational environment. Computing demographic attributes may offer a better insight about student's personality or situation, but in terms of education, it may not reflect its focus or how well the student it is engaged in the activity.

Although the dataset is useful for final grade prediction and it is used in many papers, some entirely new [2] but, it is impossible to predict the final grade at early stages and to prevent the failure. Another problem of this accessible dataset is that it does not offer a good classification accuracy for the final grade or the other good grades. For both of the datasets (mathematics and Portuguese), you will hardly get any better than 50% accuracy. Depending on the data preprocessing the analyst can obtain better results like in [3] or [4] but still the last grade (G3) depends on G1 and G2.

Another popular dataset is presented by Amrieh et. al. in [5] and previously in [6] and it's available on Kaggle platform. The data is gathered from a learning management system called Kalboard 360 using a learner activity tracker tool which is called experience API (xAPI). The xAPI tool is a component of the training and learning architecture that allows tutors to monitor the learning progress and actions like reading an article or watching a video. The authors collected a variety of features divided into four categories: demographic features (4), academic background features (6), parents' participation on the learning process (2) and behavioural features (4). Initial experiments conducted on the 480 instances dataset reveal a good accuracy which varies from 70% to 80% as the authors state in the paper. The

dataset was used in many papers with slightly better results for classification [7] or clustering [8] tasks.

There are also newer but not so popular datasets like *Open learning analytics dataset* which is described in [9] and discussed [10]. The dataset consists of several .csv files which describe tables from the database like courses, assessments, studentInfo, available materials and their relations. In this case, we do not have a dataset with a specific number of instances because it depends on which tables we want to merge, but the number of students that contributed to the dataset is huge. In studentInfo.csv file, there are 32593 recordings, each of them having a column for the final result which can be used as a class.

Another newer dataset is offered by Duolingo [11] which aims for a shared task on second language acquisition modelling [12] and they also launched a competition regarding this dataset. Regarding the competition, participants receive an English sentence and have to produce a high coverage set of translations in the target language. In order to level the playing field, the authors also provide a high-quality automatic reference translation (via Amazon), which may be considered as the baseline for the machine translation task. The data offered for the task comes from five Duolingo courses. All use English prompts, with multiple translations, although weighted by frequency from speakers of each of the following languages: Portuguese, Hungarian, Japanese, Korean and Vietnamese.

The world's largest repository of learning interaction data is PLSC Data shop [13], which offers a significant amount of learning data. In [14] the repository is well explained and in preset, there is plenty of data which can be queried and used for analysis. Still on the area of big educational data is ASSISTments Ecosystem [15] which is a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching but both of this systems it is a big difference in producing a dataset comparing to downloading one from Uci machine learning repository.

### STRUCTURE OF THE DATASET

The dataset consists of eleven attributes which describe the student activity performed by 275 students (instances) performed during tests. We consider a number of eleven attributes even if in the list in which we present them we count thirteen attributes because MeanTestGrade is the same but in two versions: with continuous values and with discretized values and the exam grade also have two versions: with exact values obtained at the exam and with two values (0 and 1) which signals failure or success.

The attributes are relevant for the activity performed during the testing period, and the data was collected in two distinct years of study: 2018 and 2019. The dataset is publicly

available on Kaggle<sup>1</sup> and can be used for further experiments.

1. *NumberOfLogins* - the number of logins made by the student during the testing period
2. *TimeSpentOnPlatform* - the total time spent on the platform by the student
3. *NumberOfTests* - the number of tests the students took
4. *TimeSpentForTests* - the total time spent by the students to take tests
5. *AverageTimePerTest* - the average time spent for taking one test
6. *NumberOfConcepts* - the total number of concepts included in his tests
7. *NumberOfActions* - the total number of actions logged for the student
8. *NumberOfRevisions* - the number of times student revised the questions from past tests
9. *LastGrade* - the grade obtained at his last test
10. *MeanTestsGrade* - the average grade obtained from all tests
11. *MeanTestsGradeD* - the same average grade but discretized with values from 4 to 10
12. *ExamGradeD* - the final grade obtained by the student at the final exam which is a discretized value
13. *PassExam* - the final grade divided into values: 0 for failure and 1 for passing the exam

Table 1. Attributes statistics summary

Attribute	Min	Max	Mean	StdDev
NumberOfLogins	1	42	7.57	5.74
TimeSpentOnPlatform	2	948	188.95	160.18
NumberOfTests	1	28	8.5	5.39
TimeSpentForTests	0	122	37.44	23.47
AverageTimePerTest	0	9.1	4.14	1.7
NumberOfConcepts	0	11	9.96	2.62
NumberOfActions	5	330	74.67	49.6
NumberOfRevisions	0	72	12.25	10.46
LastGrade	1	10	6.67	2.39
MeanTestsGrade	1.45	9.77	6.41	1.7
MeanTestsGradeD	4	10	NA	NA
ExamGradeD	3	10	6.44	1.82

Table 1 presents the attributes and a short analysis regarding their values. We focus on minimum, maximum and mean values along with standard deviation, which is presented in the last column because most of the values are numeric; these metrics are relevant in this case. Attribute 11 from the table have discrete values, and we cannot compute the mean and standard deviation, and this is why we have NA on those columns. The number from the first column corresponds to the attribute id from the previous list of items.

Each of the above-presented features is relevant for predicting the final grade and student's engagement in the learning activity. The number of logins along with time spent on platform and number of actions are influencing the grade as a quantity metric on how much the students are engaged in a learning activity through the platform as a bigger value implies more engagement. Usually a more engaged student will also be interested in gaining better grades and knowledge improving.

The number of tests is relevant for the dataset and the final grade because it is a good indicator on how good the student's implication is in the learning process and also how big is the influence of the other actions on the final result.

The number of concepts addressed by the student are relevant because they are a good indicator of how good the student's progress is. The relevance comes from the recommender system implemented in the platform, which allows the student to get question-related to a concept only after passing a certain threshold which signifies that he knows very well previous concepts.

The number of revisions refers mainly to how many times a student accessed a past test in order to see which questions were correctly answered and which was the correct answer. This concept is relevant for both learning engagement and predicting the final grade because a higher interest in the past tests means that the student aims for better results, and he wants to improve his knowledge. This feature, itself, can provide a significant insight regarding the student's behaviour because there are several cases: have a high grade and revise the questions, have a high grade but omit the revising and then the other two: have a small grade and revising and have a small grade but not revising the answers. These four situations need to be also addressed as future work as they provide valuable data which analysed can help the student's modelling and improve the grade or failure predicting.

<sup>1</sup> <https://www.kaggle.com/cristianmihaescu/dsa-test-dataset>

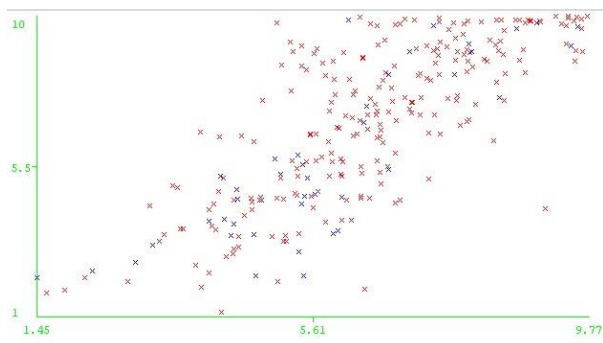


Figure 1. Last Grade vs MeanTest Grade Correlation

The last grade is relevant for the final grade or the student's level of knowledge because it marks at what level the student stopped taking tests and another motivation for computing this attribute is the correlation between it and the final result. In Figure 1, there is the correlation between *LastGrade* on the OY axis and *MeanTestGrade* on OX axis. It is visually clear that there is a correlation between these two attributes. The colour of the points for the pic corresponds to the final grade, and we have blue for fail and red for passing.

The mean grade for tests is presented in two ways, the average grade computed from all the tests and a discretized version. The grade discretization is made considering that what is more significant than 0.5 points we consider to be the next grade and what is less means the integer. For example, 4.3 will be considered as four while 4.7 will be discretized as a 5. The first version of the grade is more accurate because the value of the mark was not approximated, but it limits us to mainly regression algorithms, but discretized version allows us to use a greater variety of algorithms like classification algorithms.

Exam grade was already discretized, and it is the grade obtained by the student at the final exam. Trying to predict the exam grade based on the previous features, including the mean test grade is an excellent way to predict the student's failure. The mean test grade can also be a good estimator for the exam failure, and we aim to prevent it from making recommendations to students based on their testing results.

The motivation for adding the *PassExam* attribute can be deducted from analysing the grades distribution. Based on the grades obtained at the exam we have seven classes from which the only one is for failing the exam, and the rest of six is for passing. Dividing the data into two classes reduces the information gain split and reduces the problem to binary classification even if we must deal with imbalanced classes.

**METHODOLOGY USED FOR COLLECTING THE DATA**

The data was collected during the Data Structures course and is based on the graphs related topics. Graph topics represent half of the Data Structures course and are taught in the middle of the semester so deepening this part will have a significant impact on the final grade. Another benefit is that if we can predict the student's failure during this period, there is still enough time to catch up so building a dataset along with a system which can trigger a failure alert can be very beneficial for students who are taking this course.

The flow for constructing the dataset starts from the student who takes tests, then these tests are logged in the e-Learning platform, and after the testing period is finished, we can export the logs saved in the database and feed them to the dataset generator tool. The tool is able to execute queries on the database, computes the features for each student and build the dataset which can be used for signalling failure tot the student, to provide feedback regarding the knowledge level or to predict the exam result.

The testing period started after the graph's chapters were taught and the student has enough knowledge to answer the questions. The testing procedure is based on a concept map which is represented by a directed acyclic graph. Students had to take at least five tests from the graphs in order to have a testing grade computed and to be included in the dataset. The number of questions for each test varies from eight to ten because we logged in this dataset two years of studies and in the first year we considered ten questions per test, and in the next year, we decreased the number of questions. The reason for decreasing from ten to eight questions per tests is that in many cases, students were not able to complete the questions from the last concepts of the graph

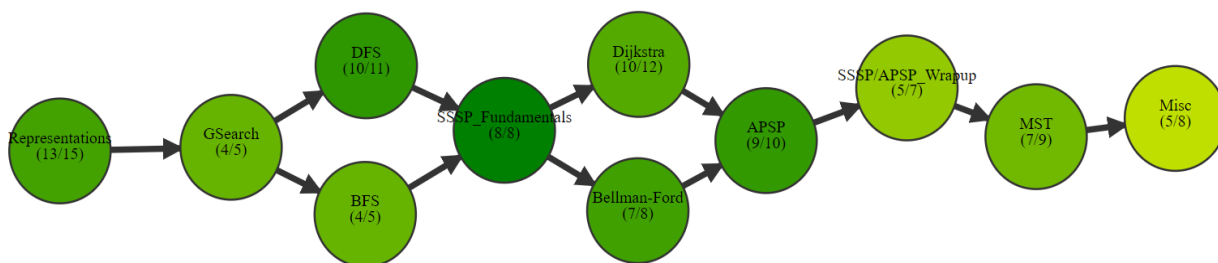


Figure 2. Graph example

Table 2. Concepts and questions per concept

Concept	No. of questions
Representations	15
GSearch	5
DFS	11
BFS	5
SSSP-Fundamentals	8
Dijkstra	12
Bellman-Ford	8
APSP	10
SSSP/APSP-Wrapup	7
MST	9
Misc	8

Table 2 presents the concepts along with the number of questions allocated for each concept. There are 98 concepts distributed for 11 concepts, and their dependency graph is presented in Figure 2. The succession of the concepts in the graph corresponds to the succession during the semester, so first they will learn *Representations*, then *GSearch* and so on. The first test which will be held will have questions from *Representations* concept and then after the student respond very good to the questions from this concept, he will advance to *GSearch*. The threshold for advancing from a concept to another is 50% and to have a concept accomplished is 75%, so a student that took tests from *GSearch* responded correctly to at least 50% of the questions form *Representations* and in order to skip getting questions from *Representations* he must have answered correctly to more than 75%.

**EXPERIMENTS**

This section presents a set of experiments conducted using this dataset. Our aim, in this case, is to explore the dataset and to investigate if we can predict the tests grade and the final result based on the logged data. A good correlation between features computed for the dataset and any of the labels will make the features we computed good predictors for the final result or the results of the test. The experiments are done using machine learning, having regression and classification tasks. For conducting experiments on this dataset, we used Weka [16] machine learning library which offers a easy to use variety of algorithms. For all the algorithms used in our experiments we considered the default values for the tuning parameters offered by Weka. Two

other kernels which can be used for further and more complex experiments are available on Kaggle<sup>2</sup>.

Table 3. Feature importance

OneRAttributeEval		InfoGainAttributeEval	
Exam	Test	Exam	Test
2	9	9	9
8	5	4	8
9	1	2	4
3	6	3	7
1	8	5	1
6	3	8	5
7	4	6	2
4	2	7	3
5	7	1	6

In order to get a better insight regarding the features and what is their importance in the dataset we use *OneRAttributeEval* and *InfoGainAttributeEval* algorithms with *Ranker* search method. There are several methods which can be used to compute the feature importance in the dataset because the feature importance depends on how we construct the model. We consider two different algorithms to analyse the dataset and see which features play a big role in the classification process. For both the algorithms we used the *MeanTestGradeD* and *Failure* attributes to evaluate the dataset so Table 3 presents on the first column the feature importance for *InfoGainAttributeEval* algorithm used on *Failure* attribute because we use *Failure* to evaluate the exam grade and on the second column the evaluation performed on *MeanTestGradeD* attribute. The same follows for the *InfoGainAttributeEval* algorithm and the values represent the attribute indices from the items list.

The attributes are included in Table 3 into descending order of their importance. Hence, the most important attribute in the dataset is the one from the first line, and on the last line, we have the least important attribute. One first conclusion when analysing the table is that the last grade obtained at the tests have a significant impact on the information gained and the number of revisions of the past tests are also important.

<sup>2</sup> <https://www.kaggle.com/cristianmihaescu/dsa-test-dataset/kernels>

**Predicting the final tests result**

First part of the experiments focus on predicting the test results and analysing the correlation between the logged data and the tests results.

*Table 4. Results obtained for regression task*

Algorithm Name	Correlation coeff	RMSE
SimpleLinearRegression	0.7719	1.0793
LinearRegression	0.782	1.0582
SMOReg	0.7805	1.0647
AdditiveRegression	0.7359	1.1631
RegressionByDiscretization	0.6875	1.2917
GaussianProcesses	0.5961	1.3816
RandomForest	0.7775	1.0704
RandomTree	0.6032	1.5254
DecisionStump	0.67	1.2604

Table 4 presents the results for a selection of algorithms used on the dataset for the regression task. On the first column of the table, we have the algorithm name as it appears in Weka, on the second column we have the correlation coefficient and on the final column we have RMSE which stands for Root Mean Squared Error metric. The best result was obtained by Linear regression with a correlation coefficient of 0.782 and an RMSE of 1.0582. For this case, we eliminated the final grade from the dataset and also the discretized version of the grade.

*Table 5. Results for classification task using DT*

Algorithm Name	Accuracy	RMSE
J48	36.36	0.36
DecisionStump	24.72	0.30
LMT	41.45	0.29
HoeffdingTree	37.45	0.33
RandomForest	40.36	0.3
RandomTree	31.63	0.4
REPTree	34.18	0.31

Table 5 presents the results obtained for classification tasks using decision trees algorithms. In this case, we used the discretized version of the final testing grade. The results, in this case, are not great because most of the features are numeric and the label is a nominal value. We included all the decision trees along with some ensembles because in most of

the cases decision trees offer comprehensible models which are valuable for analysis of educational data. In the case of the Random Forest algorithm, which was the algorithm that outperformed the others, we used 100 iterations for training.

*Table 6. Classification using various algorithms*

Algorithm Name	Accuracy	RMSE
LogisticRegression	43.63	0.29
NaiveBayes	24.72	0.30
BayesNet	37.81	0.31
AdaBoostM1	24.72	0.30
DecisionTable	36.72	0.30
ZeroR	24.72	0.32
OneR	38.9	0.39

Table 6 continues the experiments conducted with discretized values for the final testing grade, and we still cannot get good results so predicting the grade of the test based on the actions performed during the testing period is a difficult task. In both tables 5 and 6, we kept only the discretized version of the grade and eliminated the final result along with the continuous version of the testing grade.

**Predicting the student's failure**

Considering a scenario in which student finished the graph testing period and we want to prevent failure at the final exam we divided the final grade into two classes: 1 and 0: 1 for passing the exam and 0 for a possible failure. In this section, we aim to explore how most common machine learning algorithms work on this dataset for predicting the student's failure. This task is relevant for the dataset as the testing period ends before the exam period and there is still time to trigger the alarm regarding student's final result at the exam and make recommendations regarding their activity.

*Table 7. Results for predicting student's failure*

Algorithm Name	Accuracy	RMSE
J48	80.72	0.39
RandomForest	82.90	0.37
RandomTree	79.27	0.45
LMT	84.36	0.36
DecisionStump	83.27	0.37
HoeffdingTree	84.36	0.36
REPTree	83.63	0.37
ZeroR	84.36	0.36
OneR	84.36	0.39

JRip	82.18	0.37
Logistic	84.36	0.36
NaiveBayes	72.36	0.46

Table 7 summarises the results for predicting the student's failure. In the first part of the table, several decision trees algorithms are presented, and after that, a selection of several other algorithms. An accuracy of 84.36, which is also the best accuracy obtained in this case, is obtained by several algorithms and overall accuracy is above 80% correctly classified instances. This accuracy is consistent over several algorithms from different classes, and it is obtained just running the algorithms without doing any parameters tuning or applying features engineering techniques

**CONCLUSIONS AND FUTURE WORK**

In this paper, we presented a new dataset computed from testing activities performed during part of the semester at Data Structures Course. The dataset is presented and analysed in order to understand how useful it can be for predicting early student's failure. We presented several metrics for better dataset understanding and also inspected the ranking of the features using two algorithms in order to see which actions play a significant role in student's final grade at both tests and exam. Regarding this analysis, we found that the last grade obtained at the tests they took was essential and produced significant information gain.

Another focus was to evaluate how good is the dataset at predicting the tests final grade and student's failure at the final result. For this exploratory analysis, we used a selection of algorithms which covered a wide variety of situations. Regarding dataset evaluation, we obtained a good correlation for regression tasks and good accuracy for predicting the student's failure while using classification algorithms to predict the final grade was not offering good accuracy. Predicting the student's failure before the exams period is an important task, and it is useful for many learning environments as it is based on generic extracted features.

As feature work, several other algorithms are worth to be considered from both machine learning and deep learning areas. From machine learning, the gradient boosting machines are a class of algorithms that are worth exploring for better accuracy in both regression and classification tasks. Regarding deep learning algorithms, it is worth exploring neural networks that deal with numeric values. Another future work is that on the same dataset it is worth exploring other labels for predicting the tests grade or the exam's grade.

**REFERENCES**

1. Cortez, Paulo, and Alice Maria Gonçalves Silva. "Using data mining to predict secondary school student performance." (2008).

2. Al-Shehri, Huda, Amani Al-Qarni, Leena Al-Saati, Arwa Batoaq, Haifa Badukhen, Saleh Alrashed, Jamal Alhiyafi, and Sunday O. Olatunji. "Student performance prediction using support vector machine and k-nearest neighbor." In 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-4. IEEE, 2017.

3. Satyanarayana, Ashwin, and Mariusz Nuckowski. "Data mining using ensemble classifiers for improved prediction of student academic performance." (2016).

4. Al-Obeidat, Feras, Abdallah Tubashat, Anna Dillon, and Babar Shah. "Analyzing students' performance using multi-criteria classification." *Cluster Computing* 21, no. 1 (2018): 623-632.

5. Amrieh, Elaf Abu, Thair Hamtini, and Ibrahim Aljarah. "Mining educational data to predict student's academic performance using ensemble methods." *International Journal of Database Theory and Application* 9, no. 8 (2016): 119-136.

6. Amrieh, Elaf Abu, Thair Hamtini, and Ibrahim Aljarah. "Preprocessing and analyzing educational data set using X-API for improving student's performance." In 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1-5. IEEE, 2015.

7. Uzel, Vahide Nida, Sultan Sevgi Turgut, and Selma Aysel Özel. "Prediction of Students' Academic Success Using Data Mining Methods." In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1-5. IEEE, 2018.

8. Bharara, Sanyam, Sai Sabitha, and Abhay Bansal. "Application of learning analytics using clustering data Mining for Students' disposition analysis." *Education and Information Technologies* 23, no. 2 (2018): 957-984.

9. Kuzilek, Jakub, Martin Hlosta, and Zdenek Zdrahal. "Open university learning analytics dataset." *Scientific data* 4 (2017): 170171.

10. Li, Rumei, Chuantao Yin, Xiaoyan Zhang, and Bertrand David. "Online learning style modeling for course recommendation." In *Recent Developments in Intelligent Computing, Communication and Devices*, pp. 1035-1042. Springer, Singapore, 2019.

11. Settles, Burr, Chris Brust, Erin Gustafson, Masato Hagiwara, and Nitin Madnani. "Second language acquisition modeling." In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pp. 56-65. 2018.

12. Kormos, Judit. *Speech production and second language acquisition*. Routledge, 2014.

13. Koedinger, Kenneth R., Ryan Sjd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. "A data repository for the EDM community: The PSLC DataShop." *Handbook of educational data mining* 43 (2010): 43-56.

14. Stamper, John, Ken Koedinger, Ryan Sjd Baker, Alida Skogsholm, Brett Leber, Jim Rankin, and Sandy Demi.

- "PSLC DataShop: A data analysis service for the learning science community." In International Conference on Intelligent Tutoring Systems, pp. 455-455. Springer, Berlin, Heidelberg, 2010.
15. Heffernan, Neil T., and Cristina Lindquist Heffernan. "The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching." International Journal of Artificial Intelligence in Education 24, no. 4 (2014): 470-497.
16. Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter 11, no. 1 (2009): 10-18.