2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34763

Performance Prediction of Computer Science Students in Capstone Software Engineering Course Through Educational Data Mining

Dr. Saffeer Muhammad Khan, Arkansas Tech University

Saffeer M. Khan received Ph. D. degree in Electrical and Computer Engineering from the University of North Carolina at Charlotte, Charlotte, NC, USA in 2013. He is an Associate Professor in the Department of Electrical Engineering at Arkansas Tech University. His research interests include signal processing for audio and acoustics, internet of things and machine/deep learning, engineering education, and K-12 and higher education collaboration. Dr. Khan is the Chair of ASEE Midwest Section.

Dr. Mohamed Ibrahim, Arkansas Tech University

Mohamed Ibrahim, PhD Associate Professor of Curriculum and Instruction College of Education Arkansas Tech University (479) 964-0583 ext. 2452

Dr. Syed Ali Haider, State University of New York at Fredonia

Performance Prediction of Computer Science Students in Capstone Software Engineering Course through Educational Data Mining

Abstract

Educational data mining has been extensively used to predict students' performance in university courses to plan improvements in teaching and learning processes, achieve academic goals, and support timely interventions. Computer Science (CS) courses focus on promoting problem solving skills through writing of software code and developing solutions using computing technologies. Within a four-year CS curriculum, the sequencing of courses is deliberately designed so that knowledge gained in a prerequisite lower-level course is critical for success in upper-level courses. Overall, the CS curriculum prepares the students for a capstone experience in a final year Software Engineering (SE) course. The student success in SE course is dependent on skills such as requirement analysis, design, implementation, and testing gained in lower-level prerequisite courses. In this paper, we analyze grades data of 531 students in all under-graduate CS courses at a public university in the United States over a period of 8 years (2010 to 2018). Statistical analysis techniques including multiple linear regression, Pearson product-moment correlation coefficient, and paired samples t-test are used to analyze the data. The performance of students in SE course is investigated based on their grades in sequence of prerequisite courses including CS I, CS II, Data Structures and Object-oriented Programming. These prerequisite courses teach and test fundamental and advanced programming skills essential for success in SE course. The analysis shows CS II is a significant predictor of students' success in the SE course. We also investigate the relationship between study of theoretical concepts and their application by examining the correlation between CS II (theory) and Data Structures (application) courses. Results shows a strong and positive correlation between students' academic performance in the Data Structures course and CS I. We also observe the correlation between CS I and CS II. CS I builds fundamental concepts such as syntax, data types, control structures, selection statements, functions, and recursion while CS II focuses on advanced tools to use the concepts studied in CS I for problem solving. The results indicate a significant difference in mean grades in both courses.

Introduction

Knowledge discovery and data mining have been used to determine patterns in educational information system databases such as those containing admissions, registration, course records, grading and other students records and information. The pattern in the data can help schools manage their students and enhance their educational outcomes through timely and effective interventions. Using educational enterprise resource planning systems (such as Ellucian Banner), that provide important student information under one domain, the institutions of higher education collect large datasets that can provide important insights through prediction and analysis of student performance. The analysis of the data can provide a comprehensive view to help student advisors stay on top of their performance and plan and implement corrective actions. One of the key challenges in enterprise data mining is the large growth of collected data over time as the universities collect data on student's socioeconomic backgrounds and learning environments. As the volume of the data continues to increase, there is a need to discover meaningful information from the large datasets.

The university curriculum is designed to guide the students learning experiences through a deliberate sequencing of courses which are reviewed and revised regularly to stay updated with the changing needs of students and the society. This is an important step to attract and retain more students and ensure their success. College level Computer Science (CS) curriculum is an example of this approach and follows the curriculum guidelines under Criterion 5 of ABET Criteria for accrediting Computing programs. One of the strengths of CS curriculum is its scope. CS students survey fundamental science and mathematics subjects, then advance to treatments of different sub-disciplines within their chosen fields, preparing to enter a professional community of increasing breadth. The deliberate sequencing of courses is designed to reinforce and build upon connections among different subjects. In this study, we analyze the strength of these connections through analysis of relationship between the student's performance in prerequisite course and their grades in the next level course. We had access to a large dataset of 531 undergraduate CS students at a public university in the United States to analyze the contribution of CS courses on student performance in capstone course on Software Engineering. We analyzed the data to study the relationship between students' performance in Data Structures, Computer Science II and Computer Science I. The analysis techniques include multiple linear regression, correlation analysis, and paired samples t-tests. The next sections describe summary of work in educational data mining, and performance prediction and explain research questions, methodology of analysis, results, and conclusion.

Educational Data Mining

Student progress monitoring is an effective tool used by universities across the world to ensure student success. Universities generate large amounts of data every semester. This data can be used to analyze trends and patterns that can help advisors and instructors lead students to success in their degrees. Educational data can be retrieved in various levels of granularity thanks to intensive data keeping ensured by most universities. In most cases data is stored in multiple databases linked to systems provided by multiple vendors. These systems often link to the main content management system being used by the university. Moodle and Banner are examples of such content management systems.

Universities realize the importance of their data and the potential it has which can allow them to make more informed decisions [1], [2] related to recruitment and retention. Retention being on the top of the list for universities, data mining provides avenues and methodologies that can be used to extract meaningful information that can eventually benefit the students.

Educational data mining is an inter-disciplinary field that utilizes concepts from Data Mining, Machine Learning, Statistics, Pedagogical techniques, Psychology, Recommender systems [3] and visualizations [1] resulting in better understanding of student performance and in devising intervention strategies that may be used by advisors and instructors. The result is a holistic approach towards student retention and satisfaction. The techniques used in educational data mining often result in a greater appreciation of factors affecting students. According to authors in [1], these techniques include, but are not limited to Computer-supported learning analytics, Computer-supported behavioral analytics, Computer-supported visualization analytics, learning material evaluation, Computer-supported predictive analytics, self-learning behavior and social network analysis.

Performance Prediction

Computer-supported Predictive Analysis (CSPA) is extremely efficient in terms of classification rates and can be used to effectively find patterns and subsequently define learning models in educational data [1]. Student progress monitoring is a non-trivial task. Each student, as an individual, brings to the table a different set of factors affecting their performance. Anyone-size-fits-all approach will therefore lack in one area or the other. It is therefore, seen that the literature presents various options for performance measurement depending on the goals of the study.

Educational data mining has contributed significantly to student progress monitoring. Techniques such as classification and regression have been used for such predictions [4][5][6][7]. These techniques help in making educated decisions by advisors and instructors. Another important aspect of using predictive analysis is that it can be used to gauge performance of individual students which is not represented in aggregate analysis techniques such as those used in visualization analytics.

Performance prediction based on student grades have been an area of interest for researchers for many years. Various techniques have been proposed to effectively predict student performance in upcoming semester based on student performance in previous semesters [8]. Some studies also include performance in pre-university studies and SAT scores [9] while other use external information such as discussion forums [10] where students often participate.

Grades from past years can be analyzed and modeled to see significant patterns following which predictions can be made for grade with which a student may graduate [11]. Performance predictions made by the authors are biased towards students that are good at programming and not otherwise. These and various other authors have used Weka for classification [11-13]. When dealing with student grades, it is important to realize that the data is always going have missing grades. Matrix factorization is one of the techniques that is used for finding latent grades. The technique works very well for student GPA data as it does for programming predictions for Netflix users [14].

Some studies use mid-semester data to predict final exam grade [15]. Authors study data from 17 blended courses with 4989 students. One of the questions the authors tried to answer is if there is a single best way to predict student progress? Author also discuss portability of prediction models. They use LMS data (student interactions with LMS and sessional grades) generated throughout the semester to predict student performance in the final exam. Authors in [15] argue that data generated during a course should be used for analysis and prediction for timely intervention to happen and be effective.

CS Curriculum

Computer Science curriculum at the university is laid out based on specialties that are called tracks. CS major students start their coursework with a combination of programming and non-programming courses. The first programming course students take is Computer Science I (CS I) which introduces topics such as datatypes, control structures, loops, functions, file I/O and

recursion. CS I is a pre-requisite for Computer Science II (CS II) which introduces advanced concepts in object-oriented programming, inheritance, operator overloading and pointers. CS II is a pre-requisite for a whole array of advanced courses that focus various domains and are a part of various tracks. A parallel track allows students to take Visual Basic I and Visual Basic II as alternatives to CS I and CS II. Students who take Visual Basic instead of CSI and CS II, may also end up taking the same higher-level courses as students that have taken CS I and CS II.

We study student performance in the two courses CS I and CS II to investigate the relationship between grades in these two courses and advanced courses such as Object-oriented Problem Solving, Data Structures, and Capstone Software Engineering course. The analysis of student data helps us answer the research questions.

Software Engineering is an advanced course that utilizes concepts learned in foundation courses as well as 200- and 300-level courses. Students take up a term project and go through all phases of software development i.e., Requirement gathering, Design, Development, Testing and Deployment. It is for this reason that we consider Software Engineering a Capstone Course an important course for this analysis.

Research Questions

The purpose of this study was to examine students' academic performance in the computer science department based on the course completion sequence. Therefore, this study was guided by the following research questions:

Question 1: What courses in computer science major best predict students' success in the capstone software engineering course?

This primary research question was at the heart of the study, as the answer to this question will help computer science professors and administrators to identify courses that may improve students' capstone software engineering course completion rate.

Question 2: Is there a relationship between students' academic performance in the course Computer Science II and the course data structures?

Question 3: Does students' academic performance in the Computer Science II course differ from their performance in the computer science I course?

These questions attempt to assess students' academic performance in the computer science II, computer science I, and data structures courses. Specifically, the answer of these questions will inform the investigators the relationship between students' academic performance in these courses.

Methodology

This study utilized a between-subject design to investigate the students' academic performance in the computer science department. The investigators conducted multiple linear regression analysis, a Pearson product-moment correlation coefficient and paired samples t-test to answer the research questions.

Data Collection: The student grades' data was collected through the learning management system by the Office of Institutional Research at a public university in United States.

Participants: The participants in the present study were 531 undergraduate students majoring in Computer Science. English was reported as the native language of most students. The average reported age of the participants was 21 years.

Study Model Framework

This study proposed a model with four factors to predict students' success in the capstone project course. The four factors are the following courses: Computer Science I, Computer Science II, Data Structures and OO-problem Solving.

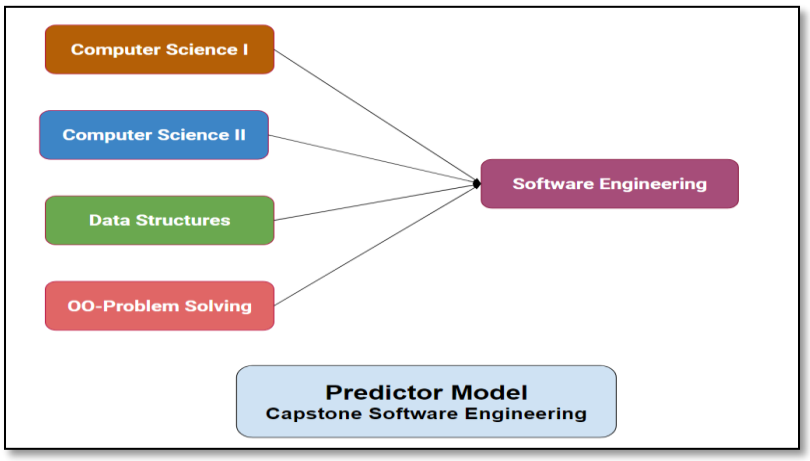

Figure 1: The proposed model to predict students' success in the capstone project course.

Data Analysis and Findings

***Question 1:* What courses in computer science major best predict students' success in the capstone software engineering course?**

Multiple Regression analysis:

Courses included in the analysis: Dependent Variable: (Software Engineering), Predictors Variables (computer science I, computer science II, data structures and OO-problem solving).

To answer this question, the investigators conducted multiple linear regression analysis to develop a model to predict students' success in the capstone course (software engineering course) based on their performance in the following courses: computer science I, computer science II, data structures and OO-problem solving.

The analysis found that all variables were included, and none of the variables was removed from the calculation. The predictor model was able to account for 27% of the variance in the dependent variable and was statistically significant at $p < .01$. Individual predictors were examined further, and the result indicated that out of the predictors variables, the only variable found to be a significant predictor that contributed to students' success in the capstone software engineering course was the computer science II course ($t = 2.108$, $p = .01$).

The results of the regression analysis assume a strong and positive causal relationship between the computer science II course and students' success in the capstone software engineering course.

Further, the analysis forecast the strong effect of the computer science II course on students' success passing in the capstone software engineering course. Finally, the analysis predicts the trends and future values of students' success in the capstone software engineering course. Model Summary and regression coefficients summarized in Table 1-3 and charts 1& 2.

Table 1

Descriptive Statistics of the multiple linear regression analysis predicting students' success in the capstone software engineering course

	Mean	Std. Deviation	N
Software Engineering	3.0967	.89628	61
Computer Science I	3.4057	.80788	61
Computer Science II	2.8828	.97805	61
Data Structures	2.7590	1.19765	61
OO-problem solving	3.1582	.89299	61

Table 2

Model summary of the multiple linear regression analysis predicting students' success in the capstone software engineering course (n=61)

					Change Statistics				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	F Change	df1	df2	Sig. F Change
1	.518[a]	.268	.216	.79351	.268	5.137	4	56	.001

Note: *a. Predictors: (Constant), OO-problem solving, Computer Science II, Computer Science I, Data Structures b. Dependent Variable: Software Engineering*

Table 3

Summary of standardized and unstandardized coefficients of the multiple linear regression analysis Dependent Variable: Students scores in Software Engineering

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations			Collinearity	
		B	Standard Error	Beta			Zero-order	Partial	Part	Tolerance	VIF
1	(Constant)	1.048	.494	.224	2.12	.03					
	CS I	.249	.149	.282	1.66	.10	.369	.218	.19	.724	1.381
	CS II	.259	.123	-.196	2.10	.03	.378	.271	.24	.728	1.374
	Data Structures	-.146	.105	.271	-1.39	.16	.152	-.184	-.16	.668	1.496
	OO-problem solving	.272	.141	.224	1.93	.05	.401	.250	.22	.663	1.508

Note: *a. Dependent Variable: Software Engineering*

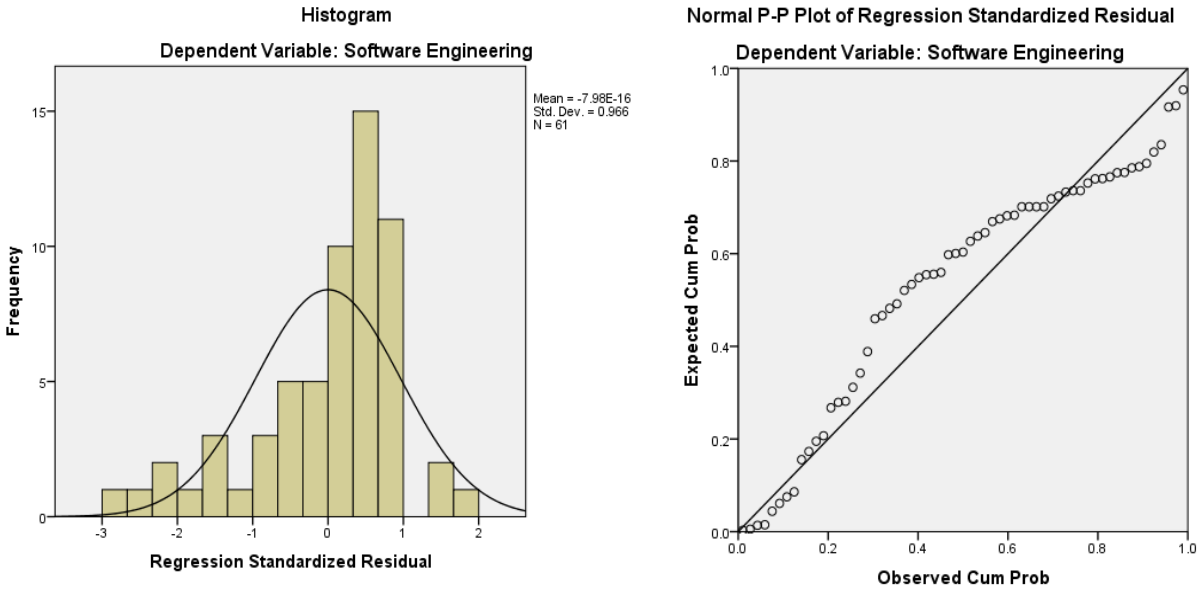

Figure 2: Histograms of regression residuals and normal probability plot.

Second question: Is there a relationship between students' academic performance in the course computer science II and the course data structures?

To answer the second question the investigators conducted a Pearson product-moment correlation coefficient to assess the relationship between students' test scores in the computer science II and the data structures courses. The analysis shows that there was a strong and positive correlation between students' test scores in the computer science II course (M = 2.6777, SD = 1.13139, n = 202), and their test scores in the data structures course (M = 2.5385, SD = 1.26379, n = 204), r = .51, p = < .001. Overall, there was a strong and positive correlation between students' test scores in these two courses and higher students' test scores in the computer science II course was associated with higher test scores in the data structures course. Further, the increase in students' test scores in the computer science II course results in a higher test scores in the data structures course. Therefore, the analysis produces a positive relationship and upward slope on a scatterplot. Table 4 and fig. 3 highlight the results of the correlation analysis for Computer Science II.

Table 4
Summarized table of correlations coefficients between students' scores in Computer Science II and Data Structures (n=141 and 202)

		Computer Science II	Data Structures
Computer Science II	Pearson Correlation	1	.507**
	Sig. (2-tailed)		.000
	Sum of Squares and Cross-products	257.290	92.358
	Covariance	1.280	.660
	N	202	141
Note: **. *Correlation is significant at the 0.01 level (2-tailed).*			

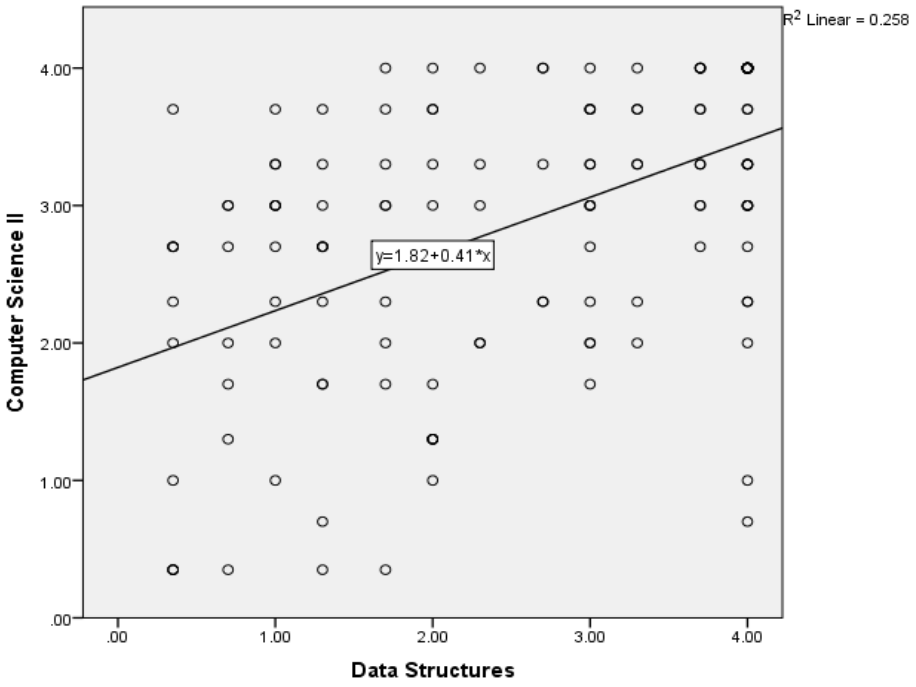

Figure 3: Correlation analysis between Data Structures and Computer Science II.

Third question: Does students' academic performance in the computer science II course differ from their performance in the computer science I course?

To answer this question, investigators conducted paired samples t-test to compare the mean of students' test scores in the computer science I and the computer science II courses. Results of the paired-samples t-test showed that there was statistically significance difference between the mean test scores of students in the computer science I course (M = 3.2490, SD = .91857, n = 146) and the mean of students' test scores in the computer science II course (M = 2.5935, SD = 1.14938, n = 146). On average, students test scores were lower in the computer science II course compared to their test scores in the computer science I course, and the mean difference was = .65548 (statistically significance). Table 5 summarize results from paired-samples t-tests performed for students in both courses.

Table 5
Paired samples t-test comparing students' scores in Computer Science I and Computer Science II (n=146)

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	Computer Science I - Computer Science II	.65548	1.19807	.09915	.45951	.85145	6.611	145	.000

Note: **. *The significant at the 0.01 level (2-tailed).*

Conclusion

In this study we have looked at mining educational data of CS students from a large public university to predict the courses that contribute to student performance in the Software Engineering capstone course in the curriculum. We also analyzed the data to understand the relationship between performance in CS II and the Data Structures and CS I and CS II. The results indicate a strong and positive causal relationship between the student performance in CS II and Capstone Software Engineering course. The results also show that students who perform well in CS II have improved performance in Capstone SE Course. The results also showed a strong and positive correlation between students' performance in CS II and Data Structures courses. Those students who did good in CS II course also performed well in the Data Structures course. The data analysis also reveals statistically significant difference between the student grades in CS I and CS II. This is in line with the coverage of topics in both courses as CS I topics are based on programming in C++ and CS II is focused on learning and mastering the course topics in Python. During this study, the team was able to perform analysis on portion of the very large database of CS students' data. The team is planning to further analyze the data to develop a more robust and comprehensive model of student performance through the CS curriculum. The team also plans to use more advanced classification and clustering techniques to answer research questions and gain insights into student performance and its dependence on sequencing and progression of CS courses.

References

[1] H. Aldowah, H. Al-Samarraie, and W. M. Fauzy, "Educational data mining and learning analytics for 21st century higher education: A review and synthesis," *Telemat. Informatics*, vol. 37, no. April 2018, pp. 13–49, 2019.

[2] C. Vialardi, J. Braver, L. Shaftr, and Á. Ortiaosa, "Recommendation in higher education using data mining techniques," *EDM'09 - Educ. Data Min. 2009 2nd Int. Conf. Educ. Data Min.*, pp. 190–199, 2009.

[3] A. Dutt and M. A. Ismail, "Logical Review on Educational Data Mining," *Int. J. Comput. Commun. Netw.*, vol. 9, no. 3, pp. 39–42, 2020.

[4] P. Golding and O. Donaldson, "Predicting academic performance," *Proc. - Front. Educ. Conf. FIE*, pp. 21–26, 2006.

[5] M. Nasiri, B. Minaei, and F. Vafaei, "Predicting GPA and academic dismissal in LMS using educational data mining: A case mining," *3rd Int. Conf. eLearning eTeaching, ICeLeT 2012*, no. Dm, pp. 53–58, 2012.

[6] H. R. M. Sweeney, J. Lester, "Next-term student grade prediction," in *2015 IEEE International Conference on Big Data (big Data)*, 2015, pp. 970–975.

[7] R. Paiva, I. I. Bittencourt, T. Tenório, P. Jaques, and S. Isotani, "What do students do on-line? Modeling students' interactions to improve their learning experience," *Comput. Human Behav.*, vol. 64, pp. 769–781, 2016.

[8] J. Xu, K. H. Moon, and M. Van Der Schaar, "A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs," *IEEE J. Sel. Top. Signal*

Process., vol. 11, no. 5, pp. 742–753, 2017.

[9] S. D. H. Hsu and J. Schombert, “Data Mining the University: College GPA Predictions from SAT Scores,” *SSRN*, vol. 35, 2010.

[10] C. Romero, M. I. López, J. M. Luna, and S. Ventura, “Predicting students’ final performance from participation in on-line discussion forums,” *Comput. Educ.*, vol. 68, pp. 458–472, 2013.

[11] M. A. Al-Barrak and M. Al-Razgan, “Predicting Students Final GPA Using Decision Trees: A Case Study,” *Int. J. Inf. Educ. Technol.*, vol. 6, no. 7, pp. 528–533, 2016.

[12] D. Kabakchieva, “Predicting student performance by using data mining methods for classification,” *Cybern. Inf. Technol.*, vol. 13, no. 1, pp. 61–72, 2013.

[13] Q. a Al-radaideh, A. Al Ananbeh, and E. M. Al-shawakfa, “CLA: Classification Model for Predicting the Suitable Study Track for School Students,” *Jordan*, vol. 8, no. August, pp. 247–252, 2011.

[14] G. Takács, I. Pilászy, B. Németh, and D. Tikk, “Matrix factorization and neighbor based algorithms for the Netflix prize problem,” in *RecSys’08: Proceedings of the 2008 ACM Conference on Recommender Systems*, 2008, pp. 267–274.

[15] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, “Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS,” *IEEE Trans. Learn. Technol.*, vol. 10, no. 1, pp. 17–29, 2017.