Analysis of Educational Robotics Activities Using a Machine Learning Approach

Lorenzo Cesaretti, Laura Screpanti, David Scaradozzi, and Eleni Mangina

Abstract This paper presents the preliminary results of using machine learning techniques to analyze educational robotics activities. An experiment was conducted with 197 secondary school students in Italy: the authors updated Lego Mindstorms EV3 programming blocks to record log files with coding sequences students had designed in teams. The activities were part of a preliminary robotics exercise. We used four machine learning techniques—logistic regression, support-vector machine (SVM), K-nearest neighbors and random forests—to predict the students’ performance, comparing a supervised approach (using twelve indicators extracted from the log files as input for the algorithms) and a mixed approach (applying a k-means algorithm to calculate the machine learning features). The results showed that the mixed approach with SVM outperformed the other techniques, and that three predominant learning styles emerged from the data mining analysis.

Keywords Educational Robotics · Educational data mining · Problem-solving process identification

L. Cesaretti (✉) · D. Scaradozzi

Department of Information Engineering (DII), Università Politecnica Delle Marche, Via Brecce Bianche, 60131 Ancona, Italy
e-mail: l.cesaretti@pm.univpm.it

L. Screpanti · D. Scaradozzi

LSIS - Umr CNRS 6168, Laboratoire Des Sciences de L’Information Et Des Systèmes, Equipe I&M (ESIL), case 925 - 163, avenue de Luminy, 13288 Marseille cedex 9, France

L. Cesaretti

TALENT Srl, via Bachelet 23, 60027 Osimo, Ancona, Italy

E. Mangina

School of Computer Science, University College Dublin, Belfield Dublin 4, Ireland

© The Author(s) 2021

D. Scaradozzi et al. (eds.), *Makers at School, Educational Robotics and Innovative Learning Environments*, Lecture Notes in Networks and Systems 240,
https://doi.org/10.1007/978-3-030-77040-2_27

203

1 Introduction

Educational Robotics (ER) is based on the constructionist learning theory proposed by Papert (1980). According to this theory, the construction of a personal and meaningful artifact promotes the construction of deep knowledge and provides a richer learning experience for students. When students design, build, program, debug and share robots, they are exploring the everyday phenomena in their lives in a new and playful way (Resnick et al. 1996). They are also improving their problem-solving and thinking skills (Scaradozzi et al. 2015, 2019b).

The spread of ER projects in schools all over the world (Miller and Nourbakhsh 2016) and the growing interest of teachers and policy-makers in this approach justify the research into new and more far-reaching ways to evaluate students' results. Researchers in the ER field have found a lack of quantitative analysis on how robotics can improve skills and increase the learning achievements of students (Benitti 2012; Alimisis 2013). For this reason, studies in recent years have focused on the evaluation of ER projects, employing qualitative (Liu 2010; Elkin et al. 2014), quantitative (Kandlhofer and Steinbauer 2016; Cesaretti et al. 2017; Scaradozzi et al. 2019a) and mixed methods (Kim et al. 2015; Chalmers 2018) to collect educational data. During a typical ER activity, students have to design a hardware or software solution to an open-ended problem and, generally, there are unlimited paths students can explore and implement. Using machine learning methods to collect and analyze ER data could help predict and classify students' behaviors, and discover latent structural patterns in large educational datasets (Berland et al. 2014).

Most studies in recent years (Berland et al. 2013; Blikstein et al. 2014; Chao 2016; Wang et al. 2017) have applied machine learning techniques to data gathered from students during programming activities without the physical presence of robots, and have obtained good results in identifying different patterns in specific coding tasks. This paper presents a research project that aims to analyze an introductory ER exercise using educational data mining techniques. The authors designed a tracking system to register log files containing the programming sequences created by (teams of) students in several secondary school classes. In this preliminary study, the collected data were used as inputs for machine learning algorithms, with two main objectives: 1—to identify different patterns in the students' problem-solving trajectories; 2—to accurately predict the final performance of the teams of students.

2 Methods

2.1 Procedure and Participants

Our experiment takes the existing research further by using educational robots. After upgrading the Lego Mindstorms EV3 programming blocks—using the Lego Mindstorms Developer kit (2019)—the authors were able to register several log files

containing the programming sequences created by 197 Italian secondary school students (organized into 56 teams), while solving an introductory Robotics exercise. Using the data collected as input for machine learning algorithms, the authors tried to identify different patterns in the students' problem-solving trajectories and predict the final performance of their teams. The experiment was conducted from March 2018 to March 2019, and involved seven Italian lower/higher secondary schools (located in the Emilia Romagna and Marche regions). The authors used convenience sampling to involve students in the experiment. Data were collected from the robotics courses organized in the schools by the startup TALENT (a partner in the research project). Two of the schools decided to offer a curricular robotics class, five decided to offer a non-curricular robotics class. 18.78% of the participants had already used robotic kits at home or at school before the experiment.

2.2 The Introductory Exercise

One of the first steps in the "Introduction to Robotics" course, whose aim was to get students involved in the experiment, was an exercise with the Lego Mindstorms EV3 motors. They had to program the robot to cover a given distance (1 m), while trying to be as precise as possible. The educator suggested a specific block to turn on the robot's motors. Using the "Move Steering" block, students can select one of the four different options shown in Fig. 1 (On for Rotations, On for Degrees, On for Seconds, On). By modifying the numeric parameter highlighted in Fig. 1, the students' team can specify how many rotations the wheels have to complete. Following the same strategy, students can decide whether to define how many degrees or for how many seconds the motors should rotate. The first two numeric parameters are for steering the robot and the speed of the motor.

Fig. 1 Lego Mindstorms EV3 "Move Steering" block

To solve this exercise, the teams had to design their solution in a maximum amount of time (15 min for upper secondary school classes, 20 min for lower secondary school classes), using no measuring tools (set squares, rulers, etc.) to check the distance covered by the robot on the floor. The measurements only applied to some of the robot's parameters (e.g., the wheel radius). The instructor gave a final evaluation for this exercise based on how accurate the team was. If the error was ≤ 4 cm, the educator considered the **challenge completed**; if the error was ≥ 4 cm the educator considered the **challenge not completed**.

2.3 Data Preparation

The tracking system collected 1113 programming sequences created by the teams of students to solve the robotics exercise described in the previous section. Several technical steps had to be performed to transform these sequences into matrices. After this transformation, the following nine indicators were calculated for each programming sequence (using a function in Python that parses the log file): **Motors** (how many motor blocks are in the sequence); **Others** (how many blocks in the sequence belong to categories other than Motors); **Added** (how many blocks have been added, compared to the previous sequence); **Deleted** (how many blocks have been deleted, compared to the previous sequence); **Changed** (how many blocks have been changed, compared to the previous sequence); **Equal** (how many blocks have stayed the same, compared to the previous sequence); **Delta Motors** (the degree of change in Motor block parameters, compared to the previous sequence, and calculated only for blocks in the "Changed" category); **Delta Others** (the degree of change in Other block parameters, compared to the previous sequence); **Trials** (how many times students activated the programming sequence on the robot).

3 Results

The authors adopted two machine learning approaches, with the aim of predicting the teams' results:

- A supervised approach, whereby the mean value and the standard deviation were calculated for each indicator presented in the previous section, and were used to create a feature matrix as the input for four machine learning algorithms (logistic regression, support-vector machine (SVM), k-nearest neighbors, and random forest classifier); the authors compared how these algorithms performed, predicting the final results of the teams of students;
- A mixed approach, which combined the benefits of both the supervised and the unsupervised methods: a k-means algorithm was applied to calculate what clusters might form from the programming sequences. Then the log files were analyzed

to calculate the percentage of sequences belonging to each cluster: these new features were then used to create a feature matrix as input for the four supervised algorithms mentioned previously.

A repeated tenfold cross validation was performed (Kim 2009) to evaluate the effectiveness of the different strategies at predicting the educational results, and four parameters were calculated (Huang and Ling 2005): Accuracy, Mean Precision, Mean Recall, Mean F1—Score. The best performance was obtained with the SVM algorithms combined with the features calculated by the k-means algorithm (the mixed approach). This combination led to the following predictions of the students' performance at resolving the exercise (challenge completed or not completed): Accuracy = 0.89 (SD = 0.002), Mean Precision 0.84 (SD = 0.007), Mean Recall = 0.84 (SD = 0.011), Mean F1 Score 0.83 (SD = 0.011).

Three problem-solving strategies emerged from the dataset collected. The “**Mathematical strategy**” groups performed a low number of total tests (compared to the other strategies), all of them with a rotation parameter equal to a fixed value (see Fig. 2 in Appendix, which shows the trend of the rotation parameter set by a group of students taken as an example for this approach). They probably used a mathematical formula to choose the robot's parameters, and never changed the value in the Motor block. The “**Incremental Tinkering approach**” groups performed a medium number of tests (compared to the other strategies); they probably used a heuristic approach to choose the robot's parameters, refined them repeatedly (see Fig. 2 in Appendix) and analyzed the feedback from the robot (for a definition of tinkering see Resnick and Rosenbaum (2013)). The “**Irregular Tinkering approach**” groups performed a high number of tests (compared to the other approaches), and their strategy is characterized by a broken line representing the number of seconds set in the Move Steering block (see Fig. 3 in Appendix), and a broken line representing the Delta Motors value calculated between two contiguous tests (which not only included changes to the number of seconds, but also modifications to the speed and steering parameters, (see Fig. 4 in Appendix). This last group probably also used a heuristic approach to choose the robot's parameters, but a certain number of large modifications could indicate difficulty understanding the feedback from the robot. These three behaviors were the most common in the dataset analyzed.

4 Conclusions

This experiment presented a number of innovative features. To the best of the authors’ knowledge, this is the first experiment involving a fair number of students (197) to gather and look at programming sequences designed by students during ER activities, and analyze them using machine learning techniques. Good results were obtained predicting the students’ performance, especially in terms of Accuracy (0.89). The three styles presented in the Results section are close to that of the “planner scientist” (similar to the “Mathematical strategy”) and the “bricoleur scientist” (similar to the “Incremental Tinkering approach” and to the “Irregular Tinkering approach”) proposed by Turkle and Papert (1992). The promising results of this preliminary study have encouraged the authors to involve new groups of students in the experiment, and also to collect data for different types of challenges, in order to validate the approach.

Appendix

See Figs. 2, 3 and 4.

Fig. 2 Graph of rotations for two groups taken as examples of the “Incremental Tinkering” and “Mathematical” approaches

Fig. 3 Graph of seconds for the group taken as an example of the “Irregular Tinkering” approach

Fig. 4 Graph of Delta Motors for the group with an “Irregular Tinkering” approach

References

- Alimisis, D.: Educational robotics: open questions and new challenges. *Themes Sci. Technol. Educ.* **6**(1), 63–71 (2013)
- Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. *Comput. Educ.* **58**(3), 978–988 (2012)
- Berland, M., Baker, R.S., Blikstein, P.: Educational data mining and learning analytics: applications to constructionist research. *Technol. Knowl. Learn.* **19**(1-2), 205–220 (2014)
- Berland, M., Martin, T., Benton, T., Petrick Smith, C., Davis, D.: Using learning analytics to understand the learning pathways of novice programmers. *J. Learn. Sci.* **22**(4), 564–599 (2013)
- Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., Koller, D.: Programming pluralism: using learning analytics to detect patterns in the learning of computer programming. *J. Learn. Sci.* **23**(4), 561–599 (2014)
- Cesaretti, L., Storti, M., Mazzieri, E., Screpanti, L., Paesani, A., Scaradozzi, D.: An innovative approach to school-work turnover programme with educational robotics. *Mondo Digitale* **16**(72), 2017–2025 (2017)
- Chalmers, C.: Robotics and computational thinking in primary school. *Int. J. Child-Comput. Interact.* **17**, 93–100 (2018)
- Chao, P.Y.: Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Comput. Educ.* **95**, 202–215 (2016)
- Elkin, M., Sullivan, A., Bers, M.U.: Implementing a robotics curriculum in an early childhood Montessori classroom. *J. Inf. Technol. Educ. Innovations Pract.* **13**, 153–169 (2014)
- Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **17**(3), 299–310 (2005)
- Kandlhofer, M., Steinbauer, G.: Evaluating the impact of educational robotics on pupils' technical- and social-skills and science related attitudes. *Robot. Auton. Syst.* **75**, 679–685 (2016)
- Kim, J.H.: Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Comput. Stat. Data Anal.* **53**(11), 3735–3745 (2009)
- Kim, C., Kim, D., Yuan, J., Hill, R.B., Doshi, P., Thai, C.N.: Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Comput. Educ.* **91**, 14–31 (2015)
- Lego Mindstorms EV3 Developer kit. Lego EV3 Mindstorms Software Developer kit official website. Retrieved from <https://education.lego.com/en-au/support/mindstorms-ev3/developer-kits> (2019)
- Liu, E.Z.F.: Early adolescents' perceptions of educational robots and learning of robotics. *Br. J. Educ. Technol.* **41**(3), E44–E47 (2010)
- Miller, D.P., Nourbakhsh, I.: Robotics for education. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, pp. 2115–2134. Springer, Cham (2016)
- Papert, S.: *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, NY (1980)
- Resnick, M., Martin, F., Sargent, R., Silverman, B.: Programmable bricks: toys to think with. *IBM Syst. J.* **35**(3.4), 443–452 (1996)
- Resnick, M., Rosenbaum, E.: Designing for tinkability. *Design, make, play: Growing the next generation of STEM innovators*, pp. 163–181 (2013)
- Scaradozzi, D., Screpanti, L., Cesaretti, L., Storti, M., Mazzieri, E.: Implementation and assessment methodologies of teachers' training courses for STEM activities. *Technol. Knowl. Learn.* **24**(2), 247–268 (2019)
- Scaradozzi, D., Screpanti, L., Cesaretti, L.: Towards a definition of educational robotics: a classification of tools, experiences and assessments. In: *Smart Learning with Educational Robotics*, pp. 63–92. Springer, Cham (2019b)
- Scaradozzi, D., Sorbi, L., Pedale, A., Valzano, M., Vergine, C.: Teaching robotics at the primary school: an innovative approach. *Procedia Soc. Behav. Sci.* **174**, 3838–3846 (2015)

- Turkle, S., Papert, S.: Epistemological pluralism and the revaluation of the concrete. *J. Math. Behav.* **11**(1), 3–33 (1992)
- Wang, L., Sy, A., Liu, L., Piech, C.: Learning to represent student knowledge on programming exercises using deep learning. In: *Proceedings of the 10th International Conference on Educational Data Mining (EDM)*, pp. 324–329. Wuhan, China (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.