Can we use Gamification to predict students' performance? A case study supported by an online judge

Filipe D. Pereira[1], Armando Toda[4], Elaine H. T. Oliveira[2], Alexandra Cristea[3], Seiji Isotani[4], Dion Laranjeira[5], Adriano Almeida[5], and Jonas Mendonça[5]

[1] Department of Computer Science, Federal University of Roraima, Boa Vista, Brazil
filipe.dwan@ufrr.br
[2] Institute of Computing, Federal University of Amazonas, Manaus, Brazil
elaine@icomp.ufam.edu.br
[3] Department of Computer Science, Durham University, Durham, United Kingdom
alexandra.i.cristea@durham.ac.uk
[4] ICMC, University of Sao Paulo, Sao Carlos, Brazil
armando.toda@usp.br,sisotani@icmc.usp.br
[5] Department of Computer Science, Estacio de Sá University, Boa Vista, Brazil
{dion.laranjeira,adriano.almeida,jonas.mendonca}@estacio.br

Abstract. The impact of gamification has been typically evaluated via self-report assessments (questionnaires, surveys, etc.). In this work, we evaluate the use of gamification elements as parameters to predict whether students are going to fail or not in a programming course. Additionally, unlike prior research, we verify how usage of gamification features can predict student performance not only as a discrete, but as a continuous measure as well, via classification and regression, respectively. Moreover, we apply our approach on two programming courses from two different universities and involve three gamification features, i.e., ranking, score, and attempts. Our results for both predictions are notable: by using data from only the first quarter of the course, we obtain 89% accuracy for the binary classification task, and explain 78% of the students' final grade variance, with a mean absolute error of 1.05, for regression. Additionally and interestingly, initial observations point also to gamification elements used in the online judge encouraging competition and collaboration. For the former, students that solved more problems, with fewer attempts, achieved higher scores and ranking. For the latter, students formed groups to generate ideas, then implemented their own solution.

Keywords: programming online judges, data-driven, data mining

1 Introduction

Gamification has been widely explored in education, and was shown to lead to positive changes in behaviour as well as better students' outcomes [4]. These outcomes vary from increasing students' engagement and motivation, to changing a specific behaviour, or improving the learning and training process [10, 11]. However, gamification typically requires a well-thought-through-design. In fact,

the literature points out that not achieving a good design may result in opposite effects, such as negatively impacting on motivation, or the emergence of undesired behaviours [14].

The majority of studies on gamification evaluate their effectveness and impact by using surveys and questionnaires [4]. Although, this approach is important and contribute to better understand the influence of gamification on users, it also may produce certain bias depending on several factors. For example, asking students to fill a survey while they are engaged in a activity may lead to demotivation, whereas asking them in the end of the activity may not give us enough information to make a conclusion since engagement and motivation are affective states that have specific triggers and do not last long. To address this issue, recent studies have focused on using real user data to attest gamification effectiveness, through the use of data mining or Machine Learning (ML) algorithms [8]. Gamification is employed to improve the learning process for specific domains, one, which we focus on here, being IT courses, specifically, programming lessons, since programming courses have a high failure rate [12, 13]. There is a lack of studies that explore and assess this theme, in general, and, in particular, the specific individual effect of promising game elements, such as badges and leaderboards. The main issue is that, in the past, gamification studies have been focused on enhancing learner motivation [8], and recently it moves towards evaluating the impact of games elements using data-driven approach. Therefore, we focus here on the following research question: *How can machine learning techniques measure the impact of gamification elements using data collected from a gamified online judge?*

To address this research question, here we evaluate a gamification design by analysing the relations between data related to gamification elements and students performance. In other words, we used a lightweight gamification-data space with three easily obtainable features[6] (ranking, score, attempts [7]) as input in a ML model with the objective of predicting students' performance. Previous research on predicting students' performance has mainly used binary classification (if students fail or not) [2, 5, 12], whereas continuous prediction of grades is considered more complex, even if more beneficial, since that allows the instructors to help students who are close to the threshold that defines success or failure. Therefore, in this work we performed both regression and binary classification to predict students outcomes.

2 Related Work

Teaching programming concepts is not a trivial task, the student needs to be motivated and engaged in order to abstract most of the concepts [5, 13]. To tackle this issue, recent studies have been using game mechanics and other concepts as a way to increase students' motivation and engagement [4].

A study conducted by Ortiz-Rojas et al. [10] applied a badge-based gamification design in an programming course environment. The authors aimed at

[6] In this work we use the term gamification feature in the same meaning of gamification element, since these elements are used as input in ML algorithms.

[7] These features were chosen due to convenience, which means they were previously implemented within the system we used in this research

evaluating students' engagement, motivation and learning performance through a quasi-experimental setting. They found that gamification had a positive effect on students' engagement; however, they could not prove that badges impacted on students' learning performance or self-efficacy. The authors explained that this may occur due to factors such as teachers' (varying) attitudes towards gamification and the (limited) length of the (experimental) course. Although promising, the study only evaluates gamification through the engagement of students.

Following, Papadakis and Kalogiannakis [11] performed a quasi-experiment to analyse if the gamification tool, ClassCraft, would be suitable to improve programming lessons in a secondary education classroom. They used this tool, alongside some digital games, as a way to manage classes with gamification concepts. They showed that ClassCraft gamification (Points, Role-playing, Badges, Progression bars, and etc.) indeed improved students' interest and attitude towards programming concepts. However, the authors did not find any significant changes in students' performance, where the control group did not outperformed the test group. The authors believed that the small sample size may have influenced the results. Although this study does present evidence on students' performance, the authors have not explored deeply how the elements in ClassCraft could impact the performance.

Finally, Denden et al. [3] presents an ongoing project which is a gamified intelligent Moodle (iMoodle) that uses learning analytics to provide dashboard for teachers to control the learning process. In this sense, [7] aimed to show the relationship between using gamification and level of performance in a MOOC on energy topics. As a result, the authors show that gamification promotes participants' engagement regardless of age, gender, or educational level. Despite the relevance of both studies, none of them analysed an e-learning environment targeted for programming students such as an online judge.

According to our related works, it seems that gamification has a tendency to improve engagement and motivation, but the literature presents mixed results towards the students' performance. Still, some studies [14, 11] states that badges and points can be used to increase the engagement of students, which is one of the reasons why we opted to analyse them in this study.

3 Methodology

To conduct our experiments, we obtained data from two different universities called, Federal University of Roraima (UnivA) and Estacio de Sa University (UnivB). UnivA classes consisted of 47 students, while UnivB of 21. We performed a longitudinal study of Programming lectures from two Computer Science courses with first year students (second semester), which took place over 4 months, synchronously, in both institutions, between March-July 2017. It is worth to mention that both courses followed the same lesson plans and lecturer.

The setup followed a blended learning method; lectures would explain face-to-face programming concepts (e.g., loops, conditional statements, arrays, recursion, abstract data type and data structures), totalising 10 topics, divided into 30 lectures. Then, students would practice those concepts in the URI Online Judge system, where part of the solved problems would count towards their final grade (10% of their final grade). Specifically, teachers assigned students 6 lists of problems, each one with 10 questions. The students were allowed to use

the programming language they were more familiar with to solve the problems, such as *Java*, *C*, *C++*, *Python* and so forth, out of 11 possible languages. All of the students registered within the system and agreed on having their data collected for academic research.

Online judge systems were chosen due their emerging popularity in education, especially for the programming domain. These tools provide automatic evaluation of students' source codes. Moreover, these systems are known to provide good support in educational contexts, due to the instant feedback for both students and teachers [5, 12]. In this work, we opted for the URI Online Judge system [1], due to its popularity, as said, among programmers. This is especially the case for the Brazilian programming community.

3.1 Approach and Gamification Features

For our research, URI gamification is based on the use of three game elements: Points, Competition and Renovation [1]. Points are represented through the score, which is accumulated when the student submits a correct answer. Competition is represented through the use of rankings in the system, these rankings aim to create a healthy competition among the students, institutions and countries. Finally, the Renovation is the element related to action of re-do a task, presented through the number of attempts that the student have to solve a problem in the system. The data was collected directly from the system's log files and is based on the users' interaction with the gamified system. Importantly, the data was collected only from the first quarter of each course, allowing for early prediction.

For a better understanding of the definitions of score, ranking, and attempts in our context, let $V_{sc} = (score_{s_1}, score_{s_2}, ..., score_{s_m})$ be a sequence of integer numbers, where $score_{s_m}$ represents the m^{th} student's score. Let the sequence $sort(V_{sc}) \subset N$ be the result of sorting V_{sc}, preserving duplicate elements. To calculate the $ranking_s$ of a student s, the position of the student s in the sequence $sort(V_{sc})$ is used. In the case of a tie, $attempts_s$ is used to break it, that is, the lower the number of attempts, the better the position in the ranking. Separately, students are ranked on a per-problem basis. In this case, the program execution time[8] decides the winner.

To test our hypotheses that gamification features are potentially good predictors for the final grade and, hence, their impact can be evaluated using a data-driven approach with ML techniques we took a wholistic approach, and computed the correlation between the latter and a wide variety of URI features [1]. To illustrate, beside the gamification features introduced in section 3.1, we analysed the frequency of answer accepted, answer with time limit exceeded, answer with compilation error, answer accepted in different categories (adhoc, data structures, graph, paradigms and etc. [1]), and others.

After analysing the Pearson and Spearman correlation between each pair of feature, we observed there are some cases of strong correlation between the original set of feature variables (multicollinearity) which can cause heteroscedasticity (the "variance level" of the residuals is not "constant") and autocorrelation (dif-

[8] If a problem is solved above a threshold time, than the feedback is the message 'time limit exceeded', and the problem is not considered solved. For more information visit: https://www.urionlinejudge.com.br/judge/en/faqs/about/judge

ferent measurement of the data that might not be independent of each other) in multivariate regression models. Thus, as a next step, we performed feature reduction. In order to analyse the importance of the features, we performed a stepwise regression (forward selection) to select the most relevant independent variables. Results pointed to the gamification elements as being the most promising: with *ranking* being the most important feature, and *score* and *attempts* following closely behind.

4 Results

Since ranking was the most relevant predictor, we decided to initially perform a deeply bivariate and univariate analysis of *ranking* and *final grades* of the students. Figure 1 shows two plots from UnivA and UnivB. For both universities, there is a strong negative correlation between those variables (*UnivA*: $r = -0.68$; *UnivB*: $r = -0.76$). The negative correlation (Figure 1) means that the better position in the ranking (lower value) a student has, the higher the grade they achieve will be (e.g., if the student is first in the ranking, this student will achieve a very high final grade). These results were interesting, especially in relation to our observations during classes, when students showed themselves to be highly *competitive* about the ranking.

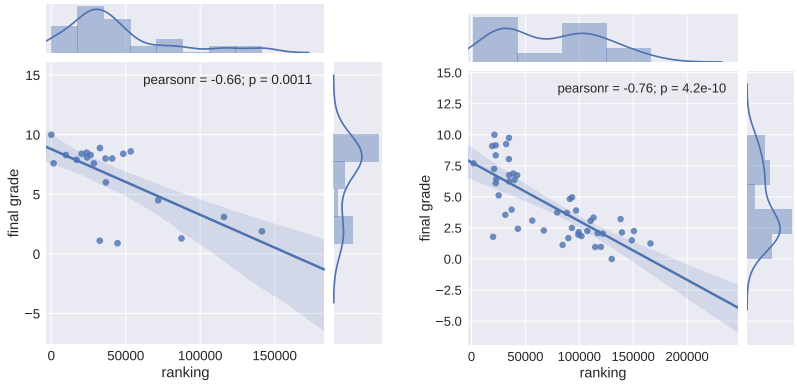

Fig. 1. Regression models, distributions of ranking and final grade of the students from each university: A, respectively, B.

Furthermore, in Figures 2 we performed a bivariate and univariate analysis of the gamification features: *attempts*, *score*, and *ranking* of students who passed and failed from UnivA and UnivB, respectively. The main diagonal of both figures shows the distribution of each aforementioned feature, where blue represents the students who passed and red, the ones who failed. On visual inspection, we can see that the distribution of *ranking* has a Gaussian shape, but this is not the case for *scores* and *attempts* (where also the passed and failed students are closer to each-other, and the curves of the first bell-shape are more similar in terms of standard deviation). Interestingly, in Figure 1 (a and b) the ranking distributions are not Gaussian (represented on the top of the plot); however, in Figure 2 (a and b), when we isolate the students who passed and failed, the

distributions are Gaussian for both passing and failing students for UnivA, and somewhat approaching a Gaussian for both type of students, for UnivB. What is similar though is that the ranking of failed students is both lower (which is expected), but also clearly with a much larger 'footprint', thus, larger standard distribution.

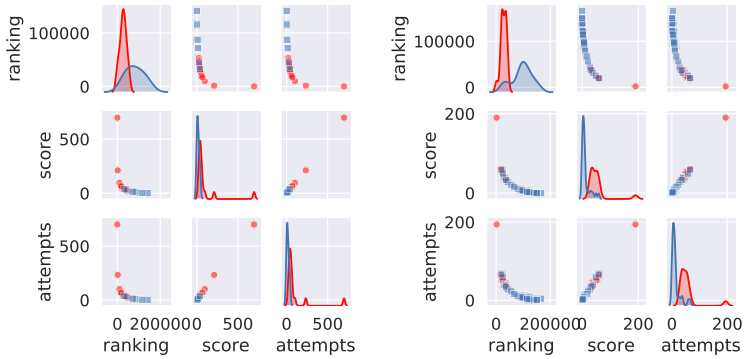

Fig. 2. Pair plot split by student final grade (pass in blue or fail in red) from UnivA (left) and UnivB (right) with univariate analysis on the plots in the main diagonal.

Additionally, we can observe in both figures (Figures 2) that students with higher grades tend to solve more problems (score), submit more code (number of attempts) and have a better rank, as expected. Another trend is that the number of attempts and score have a similar shape, which may be explained by students only committing to submit the code when they have a high level of confidence that the problem will be accepted. On one hand, that seems reasonable, since the students would not like to make many mistakes (wrong answers) to add to their 'history' trace (*attempts*), as attempts are used as a ranking criterion. Moreover, on the other hand, in the case of a submission being wrong, the URI online judge further categorises the error into different types, such as *time limit exceeded*, *compilation error*, *wrong answer* and so forth. As such, a dilemma may occur, stopping a student from submitting, unless they are really confident about the solution's correctness, when they may benefit from the feedback of the online judge during their learning curve, checking, e.g., the type of error made, which would facilitate solving the problem.

Beyond the above visual analysis, to establish if the selected gamification features can indeed work as predictors, we performed a statistical test between each gamification element and the final grade (Table 1). Indeed, there is a statistically significant difference between the students who pass and fail in terms of ranking, score and attempts, even after the Bonferroni correction ($p < 0.05/3 = 0.0167$).

Table 1. Statistical significance tests with the gamification elements and final grade.

ranking	scores	attempts
UnivAp = 2e-04 *	p = 1.9e-03 **	p = 0.0013 **
UnivBp = 7e-07 *	p = 7e-08 **	p = 1e-06 **

* T-test and ** Mann-Whitney U-test

4.1 Predictive Models

We use the $attempts_s$, $score_s$ and $ranking_s$ for each student as inputs into the classification and regression models. All algorithms (for regression and classification) were implemented using *scikit-learn* with the default settings. We did not perform hyperparameter optimisation, because of overfitting concerns, since we did not have a large dataset to separate a fold-out to perform the validation.

We tested four different structure learning algorithms on the task of predicting the student's final grade (regression), as follows: Decision Tree Regression (DT), Multivariate Linear Regression (LR), Lasso (LA), and K-Neighbours Regressor (KNN). We used them because they work well with correlated features, as in our case, as well as due to the fact that they are interpretable, i.e. can help us understand how and why each gamification element is relevant to predict the student's final grade.

To evaluate the competing methods we selected the standard Mean Absolute Error (MAE) and explained variance (r^2). The data from UnivA and UnivB was merged and split in 70% for training and 30% for testing. This merger was done to check the generalisation power of the gamification elements and to analyse if even in different educational context, as present in this cohort, we could achieve a good performance for the prediction. Notice that recent studies [12, 6, 9] argued that it is important to analyse the power of features not just in a single institution. Table 2 shows the results of the regression models.

Table 2. Results from regression algorithms.

	DT	LR	LA	KNN
MAE	1.07	1.21	1.19	1.14
r^2	0.78	0.58	0.60	0.63

DT achieved the best result, with r^2 of 0.78, which shows that a high proportion of the variance in the final grade of the students is predictable by gamification elements (ranking, score, attempts). The MAE was 1.07, which is to be interpreted as the predictive model estimation could have an error of about +/-1.07 of the student final grade (on a scale from 0 to 10). LR, LA and KNN achieved promising results as well, which demonstrate the predictive power of the gamification features, regardless of the ML algorithm.

We believe that DT was more suitable to this data, because of the nature of the method, which selects the most important feature to be the root of the subtrees, recursively using the information gain from each feature, given the entropy of the target in a given split point. We did not test Random Forest regression (or other decision tree ensembles methods such as ExtraTreesRegressor or XGBoost-Regression) since we have a light set of features (only 3) and a small database. Note that in cases of regression, these ensembles based on decision trees might create many base classifiers to calculate the average for the estimation, which is not interesting in our case, since we have features which are highly correlated with the target.

Furthermore, we applied the same features for the more traditional classification task of predicting whether the students will pass or fail. To do so, we employed the famous algorithms Random Forest (RF), Logistic Regression (LR), K-Nearest Neighbors (KNN), and Decision Tree (DT) with regularisation (max-

imum depth equal to 5 and minimum number of samples in a leaf equal to 20). Note that KNN is an instance-based algorithm, DT is a tree-based algorithm, RF tree-based ensemble algorithm, and LR is a functional algorithm. We chose those algorithms to show that the results are similar even for different classes of ML algorithms.

As a result, we achieved an accuracy ranging from 86% to 89% using cross-validation with 5 folds, as shown in Table 3. Besides that, f1-score, precision, and recall are similar for different classes, which shows that the models segregate well both classes.

Table 3. Results from classification algorithms. * result for students who failed, while ** is the results for students who passed.

	DT	LR	KNN	RF
accuracy	0.87	0.89	0.87	0.86
f1-score*	0.87	0.89	0.87	0.85
f1-score**	0.87	0.88	0.87	0.85
precision*	0.94	0.94	0.94	0.94
precision**	0.81	0.83	0.81	0.78
recall*	0.81	0.84	0.81	0.78
recall**	0.94	0.94	0.94	0.94

Summarising, our results show that we achieved competitive performance both in regression and binary classification. Prominent studies such as [6, 2], which were performed on similar educational scenarios - albeit using different databases - achieved (lower) accuracies, between [81%-85%] for the binary classification task, and [9], using a data-driven approach, explained (a somewhat higher) 85% of the variance of students final grade. Nevertheless, none of those studies used gamification elements in their methods. The closest to ours (and most recent) study, [9], not only did not use gamified elements (loosing consequently all the benefits of gamification), but we additionally employed a *lightweight feature space* (3 features), which could thus be easier to generalise to other programming courses.

Lastly, answering our research question, the ML models showed correlation between these 3 gamification elements and students performance and predicted accurately the students' outcome. This suggest that those gamification variables have a positive impact in students outcomes and, thus, our data-driven approach was useful to evaluate the gamification elements in this cohort.

4.2 Additional Discussions

Although it may seem obvious that the best ranked students with high scores will pass and the worst ranked with low scores will not, we should notice that ranking and scores are dynamic and may change from one activity to another. When tracing these features, the system could help the teacher to pay attention on behavioural changes and take preventive measures. Not so obvious and very interesting is the correlation between the number of attempts and the final grade, which may indicate student's engagement in the course. If the system traces it and detects that the student may not be enough engaged, the teacher or even the student himself may be warned about it.

Moreover, thinking about adaptive systems, recommendations can be dependent on which Gaussian the students fit in (Figure 2). If they are in the one corresponding to students who shall fail, the recommendation may be that they should solve a larger number of easy problems; this may both increase their overall ranking and help them with more of the programming basics. If they are in the second Gaussian, related to the ones who shall pass, then students could be encouraged to compete more and try to solve harder problems, so their rank within problems increases. Moreover, the use of predictors should be useful for a dynamic adaptation of the flow (i.e. the balance between challenge and reward and its progression through time).

As an interesting side-note, empirical on-site student monitoring has suggested that the students found the system motivating towards *competition*. It is to be expected that students with lower programming levels would benefit more from collaboration, whilst students with higher understanding of programming would benefit more from competing with each other. In this sense, it is worth mentioning that gamification caused some unexpected behaviour. According to the instructors of the courses, the competition generated by the system caused the students to *cooperate*, and thereby, the whole class have grown more or less equally. Whilst unexpected, this outcome was considered by the same teachers as positive. The cooperation has happened when students were trying to solve difficult programming problems. In these cases, students usually formed groups to come up with ideas and then each student tried to implement his own solution. This is an interesting result, since it goes in a different direction to the existing literature, which says that behaviours that are not expected, which happen as a result of introducing gamification, are harmful [14].

5 Conclusions

Our results demonstrate, for the first time, to the best of our knowledge, that gamification features extracted from gamified online judges used in programming classes can be used as predictors of students' success in the course (i.e. final grade). Specifically, that the *students' ranking* is highly correlated to their final grade; and our models further suggest that *scores* and *attempts* are very good predictors of the students' performance. Based on these results, besides the data-driven evaluation of the gamification elements, we can predict if students are going to succeed or fail in the course, by using the gamification data contained within an Online Judge. This information can be provided to the teacher/instructor, so that they can adapt or change their pedagogical method to aid that particular student. This kind of prediction may be a new perspective on how we can use the information provided by gamification, to support the context it is inserted into. Instead of just expecting that gamification increases learners' engagement, we also use this data to enhance their learning indirectly, by informing teachers and instructors about potential outcomes and thus intervention points. Furthermore, our work shows in a systematic, data-driven manner, that, beyond being just 'bells and whistles' for education, gamification features can be fundamental elements of an educational system, and further research in this area is desired.

Acknowledgements

This research, in accordance with Article 48 of Decree nº 6.008/2006, was funded by Samsung Electronics da Amazônia Ltda, under the terms of Federal Law nº 8.387/1991, through agreement nº 003, signed with ICOMP/UFAM.

References

1. Bez, J.L., Tonin, N.A., Rodegheri, P.R.: Uri online judge academic: A tool for algorithms and programming classes. In: Computer Science & Education (ICCSE), 2014 9th International Conference on. pp. 149–152. IEEE (2014)
2. Castro-Wunsch, K., Ahadi, A., Petersen, A.: Evaluating neural networks as a method for identifying students in need of assistance. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. pp. 111–116. ACM (2017)
3. Denden, M., Tlili, A., Essalmi, F., Jemni, M., Chang, M., Huang, R., et al.: imoodle: An intelligent gamified moodle to predict "at-risk" students using learning analytics approaches. In: Data Analytics Approaches in Educational Games and Gamification Systems, pp. 113–126. Springer (2019)
4. Dichev, C., Dicheva, D.: Gamifying education: what is known, what is believed and what remains uncertain: a critical review. International Journal of Educational Technology in Higher Education **14**(1), 9 (2017)
5. Dwan, F., Oliveira, E., Fernandes, D.: Predição de zona de aprendizagem de alunos de introdução à programação em ambientes de correção automática de código. In: Brazilian Symposium on Computers in Education. vol. 28, p. 1507 (2017)
6. Estey, A., Coady, Y.: Can interaction patterns with supplemental study tools predict outcomes in cs1? In: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. pp. 236–241. ACM (2016)
7. Flores, E.G.R., Mena, J., Montoya, M.S.R., Velarde, R.R.: The use of gamification in xmoocs about energy: Effects and predictive models for participants' learning. Australasian Journal of Educational Technology pp. 43–59 (2020)
8. Meder, M., Plumbaum, T., Albayrak, S.: A primer on data-driven gamification design. In: Proceedings of the Data-Driven Gamification Design Workshop. pp. 12–17 (2017)
9. Munson, J.P., Zitovsky, J.P.: Models for early identification of struggling novice programmers. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. pp. 699–704. ACM (2018)
10. Ortiz-Rojas, M., Chiluiza, K., Valcke, M.: Gamification in Computer Programming: Effects on Learning, Engagement, Self-Efficacy and Intrinsic Motivation. In: The 11th European Conference on Game-Based Learning ECGBL 2017. pp. 507–514 (oct 2017). https://doi.org/10.1109/EDUCON.2017.7943073
11. Papadakis, S., Kalogiannakis, M.: Using gamification for supporting an introductory programming course. the case of classcraft in a secondary education classroom. In: Interactivity, Game Creation, Design, Learning, and Innovation, pp. 366–375. Springer (2017)
12. Pereira, F., Oliveira, E., Fernandes, D., Junior, H., de Carvalho, L.S.G.: Otimização e automação da predição precoce do desempenho de alunos que utilizam juízes online: uma abordagem com algoritmo genético. In: Brazilian Symposium on Computers in Education. vol. 30, p. 1451 (2019)
13. Pereira, F.D., Oliveira, E., Cristea, A., Fernandes, D., Silva, L., Aguiar, G., Alamri, A., Alshehri, M.: Early dropout prediction for programming courses supported by online judges. In: International Conference on Artificial Intelligence in Education. pp. 67–72. Springer (2019)

14. Toda, A.M., Valle, P., Isotani, S.: The dark side of gamification: An overview of negative effects of gamification in education. In: Researcher Links Workshop: Higher Education for All. pp. 143–156. Springer (2017)