

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

Identifying Student Profiles within Online Judge Systems using Explainable Artificial Intelligence

Mrs A N RAMAMANI, T V N KONDALA RAO,

Associate Professor, S.V.K.P Dr KS Raju arts & Science College (A),Penugonda,W.G District, Andhra Pradesh , ramasrinivasu2@gmail.com PG scholar , S.V.K.P & Dr K.S.Raju Arts & Science college(A),Penugonda, W. G District, Andhra Pradesh, vijaythota7360@ gmali.com

ABSTRACT

Online Judge (OJ) systems are typically considered within programming-related courses as they yield fast and objective assessments of the code developed by the students. Such an evaluation generally provides a single decision based on a rubric, most commonly whether the submission successfully accomplished the assignment. Nevertheless, since in an educational context such information may be deemed insufficient, it would be beneficial for both the student and the instructor to receive additional feedback about the overall development of the task. This work aims to tackle this limitation by considering the further exploitation of the information gathered by the OJ and automatically inferring feedback for both the student and the instructor. More precisely, we consider the use of learning-based schemes—particularly, Multi-Instance Learning and classical Machine Learning formulations—to model student behaviour. Besides, Explainable Artificial Intelligence is contemplated to provide human-understandable feedback. The proposal has been evaluated considering a case of study comprising 2,500 submissions from roughly 90 different students from a programming-related course in a Computer Science degree. The results obtained validate the proposal: the model is capable of significantly predicting the user outcome (either passing or failing the assignment) solely based on the behavioural pattern inferred by the submissions provided to the OJ. Moreover, the proposal is able to identify prone-to-fail student groups and profiles as well as other relevant information, which eventually serves as feedback to both the student and the instructor.

INTRODUCTION

ORIGINALLY coined by [1], the term Online Judge (OJ) denotes those systems devised for the automated evaluation and grading of programming assignments, which usually take the form of online evaluation services capable of collecting source codes, compiling them, assessing their results, and computing scores based on different criteria [2]. These automated tools have been particularly considered in two

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

precise, yet related, scenarios [3]: (i) programming contests and competitions, and (ii) educational contexts in academic degrees. This work focuses on the latter scenario, in particular, on programming courses from Computer Science studies in higher education institutions.

OJ systems are successful in the education field because they overcome the main issues associated with the manual evaluation of assignments [4]: in opposition to human grading, which is deemed as a tedious and error-prone task, these tools provide immediate corrections of the submissions regardless of the number of participants. Moreover, the competitive learning framework that these schemes entail proves to benefit the success of the learning process [5].

Despite their clear advantages, OJ systems do not provide the student nor the instructor with any feedback from the actual submission apart from whether the provided code successfully accomplished the assignment [6]. However, the information gathered by the OJ system may be further exploited to enrich the educational process by automatically extracting additional insights such as student habits or patterns of behaviour related to the success (or failure) of the task. In this regard, one may resort to the so-called Educational Data Mining (EDM), a discipline meant to infer descriptive patterns and predictions from educational settings [7]. Within this discipline, Machine Learning (ML) is reported as one of the main enabling technologies due to its power and flexibility. Some success cases can be found in the work by [8], devoted to assessing the performance of the instructor; the approach by [9], aimed at predicting student grades at an early stage; or the work by [10], focused on detecting inconsistencies in peer-review assignments. In this work, we apply EDM to automatically provide feedback about the assignments, both to the student and the instructor, in the context of OJ systems for programming courses.

When an OJ is used for grading a programming assignment, there is usually a time slot in which students can perform as many submissions as they want. The final grade of a student in the activity is typically computed from the best submission. During that time slot, data usually exploited in EDM, such as grades obtained in previous activities or course attendance [9], may not be available. Moreover, other data used to predict student performance, such as socioeconomic background or academic success in other courses [11], may not be usable from an ethical point of view due to the potential biases it would introduce.

In spite of the lack of available data, it would still be desirable to be able to detect at-risk students before the assignment deadline. Thus, aided by the use of meta-information gathered from the submission process—e.g., the number of

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

code submission attempts or the date of the first submission—we devised an EDM approach with two types of outcomes: (i) the success probability of a new student, and (ii) the identification of different student profiles to provide feedback to both the instructor and the student thyself. Note that such pieces of information may be used not only to prevent inadequate student attitudes by providing the appropriate observations about the development of the task but also to properly adjust the difficulty of the different assignments, among other possible corrective actions towards the success of the course.

Since the set of code submissions made by a student somehow characterises the student profile to be estimated, the problem may be modelled as a Multi-Instance Learning (MIL) task [12]. This learning framework introduces the concept of bag, i.e., a set with an indeterminate number of instances that is assigned a single label [13]. MIL has been successfully considered in the EDM literature [14], as in the work by [15], which compares MIL against ML for predicting the student performance. In our case, each of these bags gathers the different code submissions made by each student, being labelled as either positive or negative depending on whether the student eventually passed the assessment by the OJ system.

Nevertheless, the fact that both ML and MIL strategies generally work in a black box manner hinders their application in this feedback-oriented context [16]. In this regard, the field of Explainable Artificial Intelligence (XAI) is gradually gaining attention to tackle such limitation by devising methodologies that allow humans to understand and interpret the decisions taken by a computational model [17]. However, while XAI has been largely studied in the ML field, this has not been the case in the MIL one [18].

Considering all the above, this work presents a method to identify student profiles in educational OJ systems with the aim of providing feedback to both the students and the instructors about the development of the task. More precisely, the proposal exclusively relies on the meta-information extracted from these OJ systems and considers a MIL framework to automatically infer these profiles together with XAI methods to provide interpretability about the estimated behaviours. In order to apply XAI to MIL problem, a novel policy for mapping the MIL representation to an ML one is proposed for the particular task at hand. The proposed methodology has been evaluated in a case of study comprising three academic years of a programming-related course with more than 2,500 submissions of two different assignments. For this, more than 20 learning-based strategies comprising ML, MIL, and MILto- ML mapping methods have been assessed and compared to prove the validity of the proposal. The results

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

obtained show that the proposal adequately models the user profile of the students while it also provides a remarkably precise estimator of their chances to succeed or fail in the posed task solely based on the meta-information of the OJ.

The rest of the work is organised as follows: Section II reviews the related literature to contextualise the work; Section III presents the proposed methodology; Section IV introduces the case of study examined; Section V details the experimental set-up considered; Section VI shows and discusses the results obtained; Section VII summarises the insights obtained in the work; and finally, Section VIII concludes the work and outlines future research line to address.

Online Judge (OJ) systems have revolutionized the way programming skills are evaluated and nurtured. These platforms offer automated assessment of coding tasks, providing instant feedback and fostering a competitive yet educational environment. The ability to identify student profiles within these systems is crucial for personalizing learning experiences and enhancing educational outcomes. Explainable Artificial Intelligence (XAI) plays a pivotal role in this domain by making AI's decision-making processes transparent and understandable, which is essential in educational settings to build trust and provide actionable insights.

Online Judge systems are integral to computer science education, offering platforms where students and programmers can submit code solutions for automated grading. Key features of these systems include immediate feedback, a diverse range of problems, and comprehensive user performance tracking. Notable examples of OJ systems are LeetCode, Codeforces, and HackerRank, which are widely used in both educational settings and coding competitions. Studies such as those by Pohl and Hösl (2018) and Ihantola et al. (2010) have highlighted the effectiveness of these systems in supporting programming education through their automated assessment capabilities and the rich data they provide for educational analysis.

Profiling students in OJ systems involves categorizing them based on their interactions and performance data. This profiling can be based on various factors such as skill level (beginner, intermediate, advanced), learning behavior (frequency and patterns of submissions), and engagement levels (consistent, sporadic). Understanding these profiles helps in tailoring educational content and

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

interventions to meet individual student needs. Research by Ala-Mutka (2005) and Leinonen et al. (2020) emphasizes the importance of automated assessment systems in identifying student profiles, which can significantly enhance personalized learning by providing insights into students' strengths and weaknesses.

Explainable AI is crucial in educational contexts as it provides transparency and interpretability in AI models, making their decisions understandable to humans. This is particularly important for educators who need to trust and comprehend the AI's insights to effectively use them in teaching. XAI techniques include model-agnostic methods like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations), as well as inherently interpretable models such as decision trees and rule-based systems. Key studies by Ribeiro et al. (2016) and Lundberg and Lee (2017) have demonstrated how these techniques can elucidate the workings of complex models, making them more accessible and trustworthy for educational purposes.

In educational environments, XAI can provide valuable insights into student performance, helping educators identify areas where students need more support and tailoring interventions accordingly. XAI also aids in ensuring fairness and mitigating biases in AI-driven educational tools. Research by Holstein et al. (2019) has highlighted the necessity for fairness and interpretability in educational AI systems, while Amershi and Conati (2009) have shown how combining unsupervised and supervised learning can build effective user models for exploratory learning environments. These applications underscore the potential of XAI to enhance the educational experience by making AI-driven insights more actionable and reliable.

Several case studies have illustrated the practical benefits of integrating XAI in educational OJ systems. For instance, Basu et al. (2013) developed Powergrading, a clustering approach that amplifies human effort in grading short answers by providing explainable insights into student responses. Similarly, Holstein et al. (2018) implemented XAI techniques in real-world educational settings, demonstrating the practical challenges and benefits of using XAI to improve learning outcomes. These studies highlight the importance of XAI in providing interpretable feedback, which is crucial for both educators and students to

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

understand and trust the AI's recommendations.

3. EXISTING SYSTEM

work by [19], who was the first to propose that academic computing assignments could be automatically graded, is considered the main precursor of current OJ systems. Nevertheless, their first formal definition was introduced by [1] who described them as a computer system that automatically grades programming assignments and provides some type of feedback to the students.

Regarding their practical use, the scientific literature comprises a large number of OJ proposals related, to a great extent, to academic institutions and educational environments. Some examples of such systems comprise the work by [20] with the Javaluador method for tasks in the Java programming language (it is described later in this paper), the URI system by the Universidade Regional Integrada for developing and improving general coding skills [21], the Peking University Online Judge (POJ) by [22] tailored to C++ courses, the CourseMaker one by the University of Nottingham for general programming tasks [23], the Youxue Online Judge (YOJ) [24] also for improving coding skills inspired on exercises from different programming contests, and the Sphere Online Judge (SPOJ) devised for E-Learning frameworks [25], among others.

Besides their use for educational purposes, OJ systems are also commonly considered in the context of coding competitions for solving algorithmic problems. Examples of such cases are the one used in the International Collegiate Programming Contest [26] or the UVa one considered in the Olympiads in Informatics [27].

The identification of struggling students in early course stages is deemed as a remarkably important topic in the education field as it suggests the instructor to provide additional resources to address the problem. In this sense, a large number of studies have assessed the influence of both extrinsic and intrinsic factors on the commented difficulties.

In relation to the extrinsic aspects, most of the existing literature resorts to the analysis of the socioeconomic position of the student or the marks obtained in previous courses [11]. The reader is referred to the manuscript by [28] for a thorough revision of these factors as it is out of the scope of this work. Regarding the intrinsic aspects—using information about the outcomes of the assignments carried out within a course—, the related literature comprises a large number of approaches since they typically yield considerably accurate predictions. Some representative examples include: the work by [29], which addresses this task in generic online learning platforms; that by [30] on preventive

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

failure detection in the context of the Moodle platform; the case of [31] that estimates this information relying on information gathered from clicker tests in peer-based instruction environments; and the approach by [9], who use course attendance as a predictor of academic outcome for the academic year.

Focusing on the case of programming courses, it may be checked that the most basic, yet successful, approaches rely on hand-crafted heuristics neglecting the use of OJ systems. For instance, Error Quotient [32] together with its refined version Repeated Error Density [33] perform this assessment by resorting to the syntax errors that occur during the compilation stage. The Watwin Scoring Algorithm [34] works in a similar way, but penalises students based on the time required to fix each type of error compared to that of their peers. [35] devised a scoring mechanism that takes into account more complex interactions, such as debugging or modifying syntactically correct code. A last example is the one by [36] that identifies at-risk students by means of a linear regression approach based on compilation errors and other indicators.

Advantages

(i) transparency methods, which represent the ones that directly convey the workings of the model; and

(ii) post-hoc explanations, which attempt to provide justifications about the reason why the model arrived at its predictions. This work frames on the latter case since, oppositely to transparency-based approaches, they avoid the need for individually adapting each learning-based model considered for the particular task at hand.

Proposed System

Considering all the above, this work presents a method to identify student profiles in educational OJ systems with the aim of providing feedback to both the students and the instructors about the development of the task. More precisely, the proposal exclusively relies on the meta-information extracted from these OJ systems and considers a MIL framework to automatically infer these profiles together with XAI methods to provide interpretability about the estimated behaviours. In order to apply XAI to MIL problem, a novel policy for mapping the MIL representation to an ML one is proposed for the particular task at hand. The proposed methodology has been evaluated in a case of study comprising three academic years of a programming-related course with more than 2,500 submissions of two different assignments. For this, more than 20 learning-based strategies comprising ML, MIL, and MILto- ML mapping

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

methods have been assessed and compared to prove the validity of the proposal. The results obtained show that the proposal adequately models the user profile of the students while it also provides a remarkably precise estimator of their chances to succeed or fail in the posed task solely based on the meta-information of the OJ.

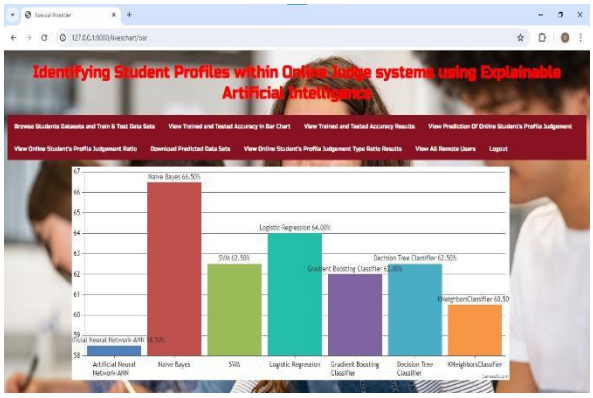
Disadvantages

• The complexity of data: Most of the existing machine learning models must be able to accurately interpret large and complex datasets to judge the Student profiles.

• Data availability: Most machine learning models require large amounts of data to create accurate predictions. If data is unavailable in sufficient quantities, then model accuracy may suffer.

• Incorrect labeling: The existing machine learning models are only as accurate as the data trained using the input dataset. If the data has been incorrectly labeled, the model cannot make accurate predictions

4. OUTPUTSCREENS



5. CONCLUSION

Online Judge (OJ) systems have been largely considered in the context of programming-related courses as they provide fast and objective assessments of the code developed and submitted by the students. Despite their clear advantages, OJ systems do not generally provide the student nor the instructor with any feedback from the actual submission besides whether the provided code successfully accomplished the assignment. While this limitation is acceptable up to some extent, it would be useful for these systems to retrieve additional pieces of information that could eventually lead to the identification of student habits, patterns of behaviour, or profiles related to the success (or failure) of the task, among others. Note that, while such types of insights are deemed as key points in the educational field, the process is not currently addressable by existing OJ-based methodologies.

This work aims to tackle this limitation by resorting to the Educational Data Mining (EDM) field. For that, the proposal considers the use of learning-based schemes from the EDM area—more precisely, Multi-Instance Labelling (MIL) and classical Machine Learning (ML) formulations—to model the student behaviour based on the code submissions provided. In addition, since these frameworks do not generally provide a human understandable feedback—

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

which is the expected output of the method—, we propose the use of Explainable Artificial Intelligence (XAI) to obtain such interpretable feedback.

This methodology has been evaluated considering a case of study with data gathered from a programming-related course in a Computer Science degree. This collection comprises the different submissions to an OJ system of two different assignments during three academic years, comprising more than 2,500 submissions from roughly 90 different students, which represents all pupils developing the commented course and approximately, 80% of the people enrolled in it. The results obtained validate the proposal: in terms of statistical significance, the model is capable of adequately predicting the user outcome (either passing or failing the assignment) solely based on the behavioural pattern inferred by the submissions provided. Moreover, the proposal is able to identify prone-tofail student groups, being hence possible to provide feedback to both the student and the instructor.

Future work considers the further validation of the model, both increasing the amount of data of the case of study as well as considering other alternative courses that also resort to OJ evaluation methods. In addition, we will consider the possibility of exploring the use of human factor characteristics drawn from, for instance, personality, self-efficacy, and motivation tests to boost the prediction accuracy of the system.

REFERENCES

[1] A. Kurnia, A. Lim, and B. Cheang, “Online judge,” Computers &
Education, vol. 36, no. 4, pp. 299–315, 2001.
[2] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal, “A
survey on online judge systems and their applications,” ACM Computing
Surveys (CSUR), vol. 51, no. 1, pp. 1–34, 2018.
[3] R. Yera and L. Mart´ınez, “A
recommendation approach for programming online judges supported by data preprocessing techniques,” Applied
Intelligence, vol. 47, no. 2, pp. 277–290, 2017.
[4] B. Cheang, A. Kurnia, A. Lim, and W.-C. Oon, “On automated grading
of programming assignments in an academic institution,” Computers &
Education, vol. 41, no. 2, pp. 121–131, 2003.
[5] L. M. Regueras, E. Verdu, M. F. Munoz, M. A. Perez, J. P. de Castro, and
M. J. Verdu, “Effects of competitive e-learning tools on higher education
students: A case study,” IEEE Transactions on Education, vol. 52, no. 2,
pp. 279–285, 2009.
[6] A. Mani, D. Venkataramani, J. Petit Silvestre, and S. Roura Ferret,

ISSN 2319-5991 www.ijerst.com
Vol. 20, Issue 2, 2024

"Better feedback for educational online judges," in Proceedings of the
6th International Conference on Computer Supported Education, Volume
2: Barcelona, Spain, 1-3 April, 2014. SciTePress, 2014, pp. 176–183.
[7] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing
undergraduate students' performance using educational data mining,"
Computers & Education, vol. 113, pp. 177–194, 2017.
[8] X. Zhang and Y. Kang, "Examining and predicting teacher professional
development by machine learning methods," in International Conference
on Neural Computing for Advanced Applications. Springer, 2021, pp.
255–269. [9] C. C. Gray and D. Perkins,
"Utilizing early engagement and machine learning to predict student outcomes,"
Computers & Education, vol. 131,
pp. 22–32, 2019.
[10] J. R. Rico-Juan, A.-J. Gallego, and J. Calvo-Zaragoza, "Automatic detection
of inconsistencies between numerical scores and textual feedback
in peer-assessment processes with machine learning," Computers &
Education, vol. 140, p. 103609, 2019.
[11] S. Alturki, N. Alturki, and H. Stuckenschmidt, "Using educational
data mining to predict students' academic performance for applying
early interventions," Journal of Information Technology Education:
Innovations in Practice, vol. 20, no. 1, pp. 121–137, 2021.
[12] J. Foulds and E. Frank, "A review of multi-instance learning assumptions,"
The knowledge engineering review, vol. 25, no. 1, pp. 1–25, 2010.
[13] M.-L. Zhang, "Generalized multi-instance learning: Problems, algorithms
and data sets," in 2009 WRI Global Congress on Intelligent
Systems, vol. 3. IEEE, 2009, pp. 539–543.
[14] S. Anupama Kumar and M. N. Vijayalakshmi, Efficiency of Multiinstance
Learning in Educational Data Mining.
Singapore: Springer,
2018, pp. 47–64.
[15] A. Zafra, C. Romero, and S. Ventura, "Multi-instance learning versus
singleinstance learning for predicting the student's performance," Handbook
of Educational Data Mining, pp. 187–200, 2010.
[16] T. Kom´arek, J. Brabec, and P. Somol, "Explainable multiple instance
learning with instance selection randomized trees," in Joint European
Conference on Machine Learning and Knowledge Discovery in
Databases. Springer, 2021, pp. 715–730.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 20, Issue 2, 2024

[17] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," Journal of Artificial Intelligence Research, vol. 70, pp. 245–317, 2021.

[18] A. B. Arrieta, N. D´ıaz-Rodr´ıguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garc´ıa, S. Gil-L´opez, D. Molina, R. Benjamins et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020.

[19] J. B. Hext and J. W. Winings, "An automatic grading scheme for simple programming exercises," Commun. ACM, vol. 12, no. 5, p. 272–275, May 1969.

[20] R. C. Carrasco, J. R. Rico-Juan, and M. A´ . Varo´, "Aprendizaje de algoritmia mediante desaf´ıos de programaci´on," in XVI Jornadas de Ense˜nanza Universitaria de la Inform´atica. Universidade de Santiago de Compostela. Escola T´ecnica Superior d'Enxe˜nar´ıa, 2010, pp. 519–522.

[21] J. L. Bez, N. A. Tonin, and P. R. Rodegheri, "Uri online judge academic: A tool for algorithms and programming classes," in 2014 9th International Conference on Computer Science Education, 2014, pp. 149–152.

[22] L. Wen-xin and G. Wei, "Peking university oneline judge and its applications," Journal of Changchun Post and Telecommunication Institute, vol. 2, 2005.

[23] C. A. Higgins, G. Gray, P. Symeonidis, and A. Tsintsifas, "Automated assessment and experiences of teaching programming," J. Educ. Resour. Comput., vol. 5, no. 3, p. 5–es, Sep. 2005.

[24] H. Sun, B. Li, and M. Jiao, "Yoj: An online judge system designed for programming courses," in 2014 9th International Conference on Computer Science Education, 2014, pp. 812–816.

[25] A. Kosowski, M. Małafiejski, and T. Noi´nski, "Application of an online judge & contester system in academic tuition," in Advances in Web Based Learning – ICWL 2007, H. Leung, F. Li, R. Lau, and Q. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelbe