IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

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

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2, 2025

https://doi.org/10.62647/ijitce.2025.v13.i2.pp486-496

EXPLAINABLE ARTIFICIAL INTELLIGENCE FOR STUDENT PROFILING IN ONLINE JUDGE ENVIRONMENTS

[1]Sajjanagandla Basava Raj, *MCA Student, Department of MCA*
[2]M G K Priyanka, MCA, (Ph.D),Assistant Professor, Department of MCA
[12]Dr KV Subba Reddy Institute of Technology, Dupadu, Kurnool

ABSTRACT

Because they provide quick and impartial evaluations of the code that students write, online judge (OJ) systems are frequently taken into consideration in programming-related courses. Based on a rubric, such an evaluation often yields a single conclusion, usually indicating whether the submission completed the task satisfactorily. Nevertheless, it would be advantageous for the student and the teacher to get more feedback regarding the task's overall progress, as such data can be considered inadequate in an educational setting. By taking into account the potential for future utilisation of the data collected by the OJ and automatically deriving feedback for both the teacher and the student, this study attempts to address this issue. More specifically, we examine the modelling of student behaviour using learning-based schemes, including Multi-Instance Learning and conventional Machine Learning formulations. Additionally, Explainable AI is being considered to give feedback that is intelligible to humans. The concept was assessed using a case study that included 2,500 entries from about 90 different students enrolled in a computer science degree course that dealt with programming. The outcomes gained support the proposal: based just on the behavioural pattern deduced from the submissions made to the OJ, the model can accurately anticipate the user outcome (passing or failing the assignment). Additionally, the proposal may pinpoint student groupings and profiles that are more likely to fail as well as other pertinent data, which ultimately provides feedback to the teacher and the student.

1. INTRODUCTION

The term Online Judge (OJ), which was first used by [1], refers to systems designed for the automated assessment and grading of programming assignments. These systems typically take the form of online evaluation services that can gather source codes, compile them, evaluate their outcomes, and calculate scores based on various criteria [2]. Two specific but related scenarios [3] have drawn special attention to these automated tools: (i) programming contests and competitions, and (ii) academic degree instructional environments. The second scenario—specifically, programming classes from computer science studies at higher education institutions—is the main emphasis of this study.

OJ systems are effective in the field of education because they solve the primary problems with manual assignment evaluation [4]: unlike human grading, which is thought to be a time-consuming and error-prone process, these tools offer instantaneous corrections of the submissions, regardless of the number of participants. Furthermore, these schemes' competitive learning framework has been shown to improve learning outcomes [5].

Despite their obvious benefits, OJ systems only reveal if the given code completed the assignment successfully [6] and neither the teacher nor the student receive any feedback from the actual submission. However, by automatically extracting new insights like student habits or patterns of behaviour connected to the task's completion (or failure), the data collected by the OJ system may be further utilised to enhance the educational process. The so-called Educational Data Mining (EDM), a field designed to extract descriptive patterns and forecasts from educational environments, may be used in this context [7]. Because of its strength and adaptability, machine learning (ML) is regarded as one of the primary enabling technologies in this field. Some examples of successful projects include the work by [8], which focusses on evaluating the instructor's performance; the method by [9], which aims to anticipate student grades early on; and the work by [10], which focusses on identifying discrepancies in peer-reviewed assignments. In this study, we use EDM to automatically give the teacher and students feedback on the assignments in the context of OJ systems for programming classes.

There is often a window of time within which students can submit as many programming assignments as they like when an OJ is used for grading. The best submission is usually used to determine a student's final mark for the task. Data that is often exploited in EDM, such grades from prior assignments or attendance in class [9], could not be accessible during that time frame. Furthermore, because of the possible biases they can bring, additional variables used to predict student performance—like socioeconomic background or academic accomplishment in other courses [11]—would not be useful from an ethical standpoint.

In spite of the lack of accessible data, it would still be beneficial to be able to recognise at-risk pupils before the assignment deadline. We therefore developed an EDM approach with two types of outcomes: (i) the success probability of a new student, and (ii) the identification of various student profiles to provide feedback to both the instructor and the student themselves. This was made possible with the help of meta-information obtained from the submission process, such as the number of code submission attempts or the date of the first submission. Keep in mind that by offering pertinent observations regarding the task's progress, such data can be utilised not only to prevent poor student attitudes but also to appropriately modify the level of difficulty of the various assignments, among other potential remedial measures to ensure the course's success.

The problem may be represented as a Multi-Instance Learning (MIL) job as the student profile to be estimated is somehow characterised by the collection of code contributions made by the student [12]. The idea of a bag—a collection with an arbitrary number of examples that are given a single label—is introduced in this learning framework [13]. As demonstrated in the study of [15], which contrasts MIL with ML for student performance prediction, MIL has been effectively taken into account in the EDM literature [14]. In our instance, each of these bags collects the many code contributions made by every student, and the OJ system labels them as either positive or negative based on whether the student ultimately passed the exam.

However, the implementation of both ML and MIL techniques in this feedback-oriented setting is hampered by the fact that they often operate in a black box fashion [16]. In order to overcome this constraint, the topic of Explainable Artificial Intelligence (XAI) is progressively gaining interest by developing techniques that enable people to comprehend and interpret the choices made by a computer model [17]. Nevertheless, XAI has not received the same level of attention in the MIL sector as it has in the ML one [18].

In light of everything said above, this study offers a way to determine student profiles in educational OJ systems in order to give instructors and students feedback on how the job is progressing. More specifically, the proposal considers a MIL framework to automatically infer these profiles and XAI approaches to give interpretability for the predicted behaviours. It just depends on the meta-information that has been retrieved from these OJ systems. A innovative policy for mapping the MIL representation to an ML one is suggested for the specific job at hand in order to apply XAI to the MIL problem. A case study including three academic years of a programming-related course with over 2,500 submissions of two distinct assignments was used to assess the suggested technique. To demonstrate the validity of the approach, over 20 learning-based methodologies, including ML, MIL, and MILto-ML mapping techniques, have been evaluated and contrasted. The findings demonstrate that the proposal accurately models the students' user profiles and offers an exceptionally accurate estimator of their likelihood of passing or failing the given assignment based just on the OJ's meta-information.

The remaining tasks are arranged as follows: In order to contextualise the work, Section II examines relevant literature; Section III provides the suggested approach; Section IV gives the case of research under consideration; and Section V describes the experimental setup taken into consideration. The results are presented and discussed in Section VI. The findings from the investigation are summarised in Section VII, and the work is concluded and future research directions are outlined in Section VIII.

2. LITERATURE SURVEY

A. Kurnia, A. Lim, and B. Cheang, "Online judge," Computers & Education

The online judge, a system we developed for automatically assessing programming assignments, is described and its usefulness assessed in this study. We contrasted this with the present manual grading system and demonstrated that, with proper implementation, the automatic grading method is more secure, equitable, and convenient than the manual one. The technology has been successfully tested on two courses. However, further research is required to increase the efficacy of learning with this approach.

S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal, "A survey on online judge systems and their applications

Online judges are programs made to accurately assess user-submitted algorithm source code, which is then generated and evaluated in a uniform setting. Online judges are growing in popularity for a number of reasons. As a result, we want to examine the current situation of these systems. According to their main goals, we divide them into systems that help with the organisation of programming competitions, improve hiring and education procedures, make data mining problems easier to solve, and include online compilers and development platforms as part of other unique systems. Additionally, we provide a formal description of an online judge system and provide an overview of the standard assessment techniques that these systems allow. As a last example of an online judge system that has been suggested for the resolution of challenging optimisation issues, we quickly go over the Optil.io platform. We also examine the outcomes of competitions held on this site. The contest demonstrated that online judging methods, reinforced by crowdsourcing principles, may be effectively used to precisely and effectively address challenging scientific and industrial problems.

.

R. Yera and L. Mart´ınez, "A recommendation approach for programming online judges supported by data preprocessing techniques

These days, it's usual practice to employ programming online judges (POJ) to help students learn programming since these programs provide a vast number of programming tasks that students must complete. In addition to offering challenges, a POJ automates the compilation and assessment of code. Information overload is a typical issue that students encounter when using POJ since there are so many issues available that it might be difficult to choose which one to tackle. The absence of essential information in Intelligent Tutoring Systems (ITSs) makes it difficult to integrate existing POJs into e-learning platforms. Therefore, this paper's goal is to help students who are struggling with information overload by employing a collaborative filtering suggestion strategy that eliminates programming issues that are inappropriate for their level of programming expertise. In order to compute tighter neighbourhoods and provide a more accurate suggestion, it makes use of an enhanced user-problem matrix, which suggests a better depiction of student roles. The recommendation approach also incorporates a unique data pretreatment phase that controls unusual user behaviours that can influence the suggestion generation. A POJ actual dataset is used for a case study, which demonstrates how the proposal works better than other earlier methods.

https://doi.org/10.62647/ijitce.2025.v13.i2.pp486-496

B. Cheang, A. Kurnia, A. Lim, and W.-C. Oon, "On automated grading of programming assignments in an academic institution

One of the most crucial phases in mastering computer programming is practice. Unfortunately, the enormous enrolment of many programming courses makes human grading of programming assignments a laborious and error-prone operation. Because of this, students in these classes typically receive fewer programming assignments than they should. Automating the grading process so that students may electronically turn in their programming projects and get immediate feedback is one way to solve this issue. This study examines how the Online Judge, an automated grading system, was implemented in the School of Computing at the National University of Singapore for a required first-year course teaching fundamental programming skills to more than 700 students. It details the responses and behaviour of the students as well as the challenges they faced. Additionally, the Online Judge was effectively used for both an introductory high school course and an advanced degree course. All rights reserved. Copyright 2003 Elsevier Ltd.

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

In recent years, information and communication technology (ICT) has been used in the majority of efforts to implement active learning approaches in the classroom. To capitalise on the flexibility offered by ICT, many of these initiatives have been focused on collaborative scenarios employed in remote, hybrid, or in-person interactions. In addition to the benefits of collaborative virtual environments, competitive learning may also have some beneficial impacts on the learning process. This study investigates how competitive learning affects telecom students' academic performance and level of satisfaction. The undergraduate course "Communications Networks" has made use of QUEST, an active and competitive learning tool. There are noteworthy findings about the impact of competitive e-learning resources on university students' performance and contentment.

3. SYSTEM ANALYSIS AND DESIGN

EXISTING SYSTEM

The primary forerunner of modern OJ systems is said to have been the work of [19], who was the first to suggest that academic computing assignments may be automatically evaluated. However, [1] provided the first official description of them, characterising them as a computer system that automatically assesses programming assignments and gives students feedback of some kind.

In terms of their actual application, there are several OJ ideas in the scientific literature that are mostly connected to educational settings and academic organisations. The URI system by 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) developed for E-Learning frameworks [25] are a few examples of such systems.

OJ systems are frequently taken into consideration for algorithmic problem solving in coding contests, in addition to their application in teaching. Examples of such examples include the UVa case that was taken into consideration for the Olympiads in Informatics [27] and the one utilised in the International Collegiate Programming Contest [26].

Early course identification of difficult students is considered a very significant topic in the world of education since it indicates that the teacher should offer more resources to solve the issue. In this regard, several research have evaluated the impact of both internal and extrinsic variables on the challenges mentioned.

Regarding the extrinsic factors, the majority of the studies currently in publication analyses the student's socioeconomic status or grades from prior coursework [11]. Since a detailed review of these elements is outside the purview of this study, the reader is directed to the manuscript by [28]. Since they usually produce very accurate predictions, the relevant literature includes a wide range of methodologies with regard to the intrinsic aspects—using data on the results of the assignments completed inside a course. The work by [29], which tackles this task in generic online learning platforms; the work by [30] on preventive failure detection in the context of the Moodle platform; the case of [31], which depends on data collected from clicker tests in peer-based instruction environments to estimate this information; and the method by [9], which uses course attendance as a predictor of academic outcome for the academic year, are some representative examples.

Using the example of programming classes, it can be seen that the simplest, yet most effective, methods focus on manually created heuristics while ignoring the usage of OJ systems. For example, Error Quotient [32] and its improved variant, Repeated Error Density [33], use syntax mistakes that arise during compilation to carry out this evaluation. Similar principles are used by the Watwin Scoring Algorithm [34], which penalises pupils according to how long it takes them to correct each kind of error in comparison to their peers. developed a scoring system that considers more intricate interactions, including fixing syntactically valid code or debugging it. The final example is the one by [36], which uses a linear regression technique based on compilation mistakes and other factors to identify pupils who are at danger.

Disadvantages

- Data complexity: In order to evaluate student profiles, the majority of machine learning models now in use need to be able to precisely comprehend sizable and intricate datasets.

https://doi.org/10.62647/ijitce.2025.v13.i2.pp486-496

- Data availability: In order to provide precise predictions, the majority of machine learning models need a lot of data. The accuracy of the model may degrade if data is not accessible in large enough amounts.
- Inaccurate labelling: The accuracy of the machine learning models that are now in use depends on how well the input dataset was used for training. Inaccurate labelling of the data prevents the model from producing reliable predictions.

PROPOSED SYSTEM

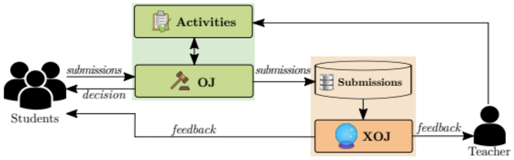
In light of everything said above, this study offers a way to determine student profiles in educational OJ systems in order to give instructors and students feedback on how the job is progressing. More specifically, the proposal considers a MIL framework to automatically infer these profiles and XAI approaches to give interpretability for the predicted behaviours. It just depends on the meta-information that has been retrieved from these OJ systems. A innovative policy for mapping the MIL representation to an ML one is suggested for the specific job at hand in order to apply XAI to the MIL problem. A case study including three academic years of a programming-related course with over 2,500 submissions of two distinct assignments was used to assess the suggested technique. To demonstrate the validity of the approach, over 20 learning-based methodologies, including ML, MIL, and MILto-ML mapping techniques, have been evaluated and contrasted. The findings demonstrate that the proposal accurately models the students' user profiles and offers an exceptionally accurate estimator of their likelihood of passing or failing the given assignment based just on the OJ's meta-information.

Advantages

(i) Transparency techniques, which are those that explain the model's operation directly; and

(ii) Post-hoc explanations, which make an effort to offer explanations for how the model came to its conclusions. This study focusses on the latter scenario since, in contrast to transparency-based techniques, it eliminates the need to modify each learning-based model separately for the specific job at hand.

4. SYSTEM ARCHITECTURE



5. IMPLEMENTATION

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Browse Students Datasets and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Online Student's Profile judgement, View Evaluation Ratio of Online

Students' Profiles, View Online Student Profile Judgement Type Ratio Results, Download Predicted Data Sets, and View All Remote Users.

View and Authorize Users

The administrator may see a list of all registered users in this module. Here, the administrator may see the user's information, like name, email, and address, and they can also grant the user permissions.

Remote User

A total of n users are present in this module. Before beginning any actions, the user needs register. Following registration, the user's information will be entered into the database. Following a successful registration, he must use his password and authorised user name to log in. Following a successful login, the user may perform tasks including registering and logging in, predicting the kind of student profile detection, and seeing their profile.

ALGORITHIM

Gradient boosting

Among other machine learning applications, gradient boosting is utilised for classification and regression problems. An ensemble of weak prediction models, usually decision trees, is what it provides as a prediction model. [1] [2] The resultant technique, known as gradient-boosted trees, often performs better than random forest when a decision tree is the weak learner. Like other boosting techniques, a gradient-boosted trees model is constructed step-by-step; however, it goes one step further by permitting optimisation of an arbitrary differentiable loss function.

Logistic regression Classifiers

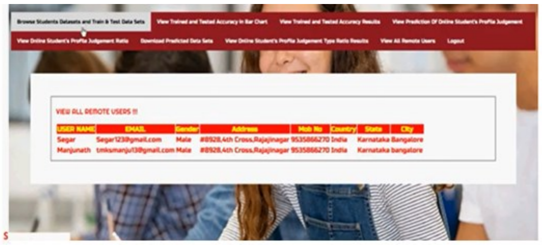
The relationship between a collection of independent (explanatory) factors and a categorical dependent variable is examined using logistic regression analysis. When the dependent variable simply has two values, like 0 and 1 or Yes and No, the term logistic regression is applied. When the dependent variable contains three or more distinct values, such as married, single, divorced, or widowed, the technique is sometimes referred to as multinomial logistic regression. While the dependent variable's data type differs from multiple regression's, the procedure's practical application is comparable.

When it comes to categorical-response variable analysis, logistic regression and discriminant analysis are competitors. Compared to discriminant analysis, many statisticians believe that logistic regression is more flexible and appropriate for modelling the majority of scenarios. This is due to the fact that, unlike discriminant analysis, logistic regression does not presume that the independent variables are regularly distributed.

Both binary and multinomial logistic regression are calculated by this program for both category and numerical independent variables. Along with the regression equation, it provides information on likelihood, deviance, odds ratios, confidence limits, and quality of fit. It does a thorough residual analysis that includes diagnostic residual plots and reports. In order to find the optimal regression model with the fewest independent variables, it might conduct an independent variable subset selection search. It offers ROC curves and confidence intervals on expected values to assist in identifying the optimal classification cutoff point. By automatically identifying rows that are not utilised throughout the study, it enables you to confirm your findings.

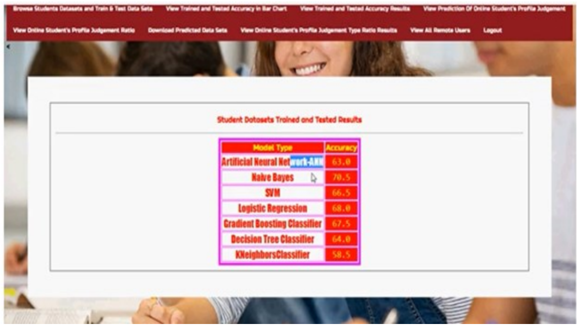
6. SCREEN SHOTS

Home | Remote User | Service Provider
udent profile identification, Online Judge systems, Multi-Instance Learning, eXplainable Artificial Intelligence, Machine


Browse Students Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Online Student's Profile Judgement
View Online Student's Profile Judgement Ratio
Download Predicted Data Sets
View Online Student's Profile Judgement Type Ratio Results
View All Remote Users
Logout

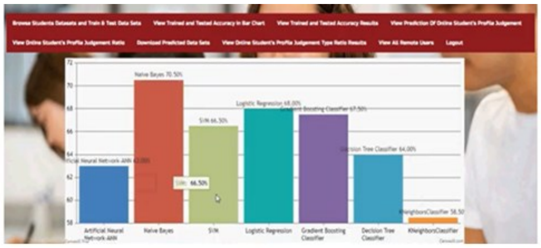
VIEW ALL REMOTE USERS !!!

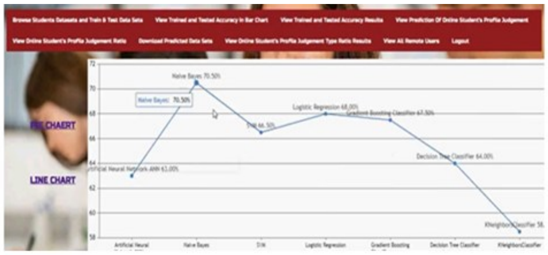
USER NAME	EMAIL	Gender	Address	Mob No	Country	State	City
Segar	Segar123@gmail.com	Male	#8928,4th Cross,Rajajinagar	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	Male	#8928,4th Cross,Rajajinagar	9535866270	India	Karnataka	bangalore

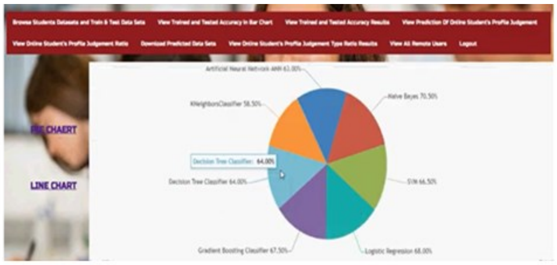

Browse Students Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Online Student's Profile Judgement
View Online Student's Profile Judgement Ratio
Download Predicted Data Sets
View Online Student's Profile Judgement Type Ratio Results
View All Remote Users
Logout

Student Dotosets Troined ond Tested Results

Model Type	Accuracy
Artificial Neural Network-ANN	63.0
Naive Bayes	70.5
SVM	66.5
Logistic Regression	68.0
Gradient Boosting Classifier	67.5
Decision Tree Classifier	64.0
KNeighborsClassifier	58.5







7. CONCLUSION

Because they offer quick and impartial evaluations of the code that students write and submit, online judge (OJ) systems have received a lot of attention in the context of programming-related courses. Notwithstanding their obvious benefits, OJ systems often just tell the teacher and student whether the given code completed the assignment satisfactorily. Although this restriction is somewhat acceptable, it would be helpful for these systems to recover more data that might eventually be used to identify student habits, behavioural patterns, or profiles associated with task success (or failure), among other things. Although these

https://doi.org/10.62647/ijitce.2025.v13.i2.pp486-496

kinds of insights are considered important in the realm of education, it should be noted that current OJ-based techniques are unable to handle the process.

Using the discipline of Educational Data Mining (EDM), this work attempts to address this restriction. In order to do this, the proposal takes into account modelling student behaviour using code submissions and learning-based approaches from the EDM field, including Multi-Instance Labelling (MIL) and traditional Machine Learning (ML) formulations. Furthermore, we suggest using Explainable Artificial Intelligence (XAI) to provide interpretable feedback, as these frameworks typically fail to deliver the method's desired result of human-understandable input.

A case study using data collected from a computer science degree course on programming has been used to assess this technique. With over 2,500 submissions from about 90 different students—representing all students taking the commented course and about 80% of the total enrollment—this collection includes the various submissions to an OJ system of two distinct assignments over the course of three academic years. The findings support the proposal: based just on the behavioural pattern deduced from the submitted work, the model can accurately predict the user outcome (passing or failing the assignment) in terms of statistical significance. Additionally, by identifying student groups that are more likely to fail, the idea makes it feasible to give feedback to both the teacher and the student.

In the future, the model will be further validated by expanding the case study's data set and taking into account additional courses that similarly use OJ assessment techniques. Additionally, in order to improve the system's prediction accuracy, we will examine the potential for investigating the use of human factor characteristics derived from, for example, personality, self-efficacy, and motivation tests.

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, “Better feedback for educational online judges,” in

https://doi.org/10.62647/ijitce.2025.v13.i2.pp486-496

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.

[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.