International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

DECODING LEARNERS: EXPLAINABLE AI FOR INTELLIGENT STUDENT PROFILING IN ONLINE CODING PLATFORMS

[1] L.Priyanka,[2] Lola Leena Sri
[1]Associate Professor, [2]MCA Student
Department Of MCA
Sree Chaitanya College Of Engineering, Karimnagar

ABSTRACT

Online Judge Systems (OJS) have become essential in assessing programming skills by automatically evaluating students' code submissions. However, most existing systems focus solely on quantitative metrics such as accuracy, execution time, and memory usage—offering limited insights into students' learning behaviors. This paper presents "Decoding Learners," an Explainable Artificial Intelligence (XAI)-based framework designed to profile students intelligently within online coding platforms. By leveraging interpretable machine learning models such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), the proposed system identifies behavioral patterns, skill gaps, and learning progress with transparency and fairness. The framework enhances educators' ability to understand students' coding approaches, decision-making patterns, and problem-solving efficiency. Experimental results show that integrating XAI methods enables better interpretability of predictive outcomes, thereby supporting personalized feedback and adaptive learning strategies.

Received: 23-09-2025 Accepted: 28-10-2025 Published: 04-11-2025

1. INTRODUCTION

In recent years, Online Judge Systems (OJS) have transformed how computer programming education is delivered and assessed. These platforms—such as HackerRank, Codeforces, LeetCode, and university-specific judges—enable students to practice coding problems, submit solutions, and receive instant feedback based on correctness, execution time, and memory efficiency. With their ability to automatically evaluate thousands of submissions in real time, they have become indispensable in programming courses, coding competitions, and large-scale computer science education.

However, while these platforms successfully automate quantitative evaluation, they fall short in delivering qualitative insights about how students learn, reason, and evolve as programmers. The metrics traditionally used—such as the number of problems solved or average accuracy—only scratch the surface of a student's learning process. They do not reveal how a student approaches problem-solving, how their coding strategies mature over time, or why certain students stagnate while others excel. This gap has motivated researchers to explore the integration of Artificial Intelligence (AI) and Learning Analytics into OJS environments.

AI-driven models have been applied to predict student performance, cluster learners, and detect at-risk students. For instance, supervised learning algorithms can predict whether a student will solve a problem based on their past attempts. However, these predictive models typically operate as "black boxes"—they provide results (e.g., "student is a high performer") without explaining why the model reached that conclusion. This lack of transparency poses a significant limitation, especially in educational contexts where interpretability and fairness are crucial. Educators need not only accurate predictions but also understandable reasoning to trust and act on AI-generated insights.

This challenge has given rise to Explainable Artificial Intelligence (XAI), a new paradigm focused on making AI decisions transparent, interpretable, and trustworthy. XAI techniques aim to bridge the gap between complex model performance and human comprehension by offering explanations that clarify how input features influence outputs. In educational data mining, explainability is particularly valuable because teachers and learners can use AI-generated explanations to adapt instruction, refine strategies, and promote self-reflection.

The proposed research, titled “Decoding Learners: Explainable AI for Intelligent Student Profiling in Online Coding Platforms,” introduces a framework that integrates XAI with online learning analytics to create interpretable student profiles. Instead of merely predicting scores or labeling students as “high” or “low” performers, this system explains how each student interacts with coding challenges and why certain behaviors correspond to particular learning patterns.

The Decoding Learners framework collects detailed data from OJS environments—such as submission timestamps, number of attempts, code efficiency, syntax errors, logical accuracy, and time taken to fix errors. Using machine learning classifiers such as Random Forests, Gradient Boosting Machines (GBM), and XGBoost, the model identifies hidden patterns in the data that correlate with different learning behaviors. The explainability layer, implemented through SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), provides transparent insights into which variables most influence student categorization.

For instance, SHAP can show that a student’s profile as a “rapid learner” is heavily influenced by reduced compile-time errors and faster problem-solving turnaround, while another student’s label as a “persistent improver” may stem from frequent resubmissions and gradual improvement trends. Educators can visualize these explanations through intuitive dashboards, thereby gaining a holistic understanding of their students’ learning journeys.

The concept of student profiling in this context refers to building dynamic, interpretable models of learner behavior. The profiles are not static grades but evolving indicators of how students think, learn, and solve problems. Examples of such profiles include:

- Consistent Performer: Demonstrates steady accuracy and low error variance across tasks.
- Rapid Learner: Shows a steep improvement curve within a short time span.
- Trial-Based Learner: Submits multiple attempts before success, indicating persistence but potential gaps in conceptual understanding.
- Conceptual Learner: Solves fewer problems but achieves high efficiency and code quality.

Through explainable AI, these profiles are interpretable and actionable. Educators can identify at-risk learners before performance declines, while students receive personalized feedback that highlights strengths and improvement areas. Moreover, the system’s transparency ensures fairness—students can see the rationale behind their categorization, addressing the ethical concerns associated with opaque AI-driven evaluation.

The increasing adoption of personalized learning environments demands systems that adapt not only to performance metrics but also to individual learning styles. The proposed framework addresses this demand by aligning AI-driven analytics with pedagogical principles. It encourages data-informed teaching, where instructors adjust course design and assessment strategies based on real-time learner insights.

From a broader perspective, “Decoding Learners” contributes to the field of Educational

Data Mining (EDM) and Learning Analytics (LA) by demonstrating how XAI can enhance both interpretability and educational value. While traditional AI systems focus primarily on optimizing predictive accuracy, the integration of XAI prioritizes understanding and trustworthiness—factors essential for real-world adoption in academic settings.

2. LITERATURE SURVEY

The integration of Artificial Intelligence (AI) and Learning Analytics in education has been an evolving research area for over a decade. Anderson and Dron (2016) explored how learning analytics can be applied in online education to improve student outcomes, emphasizing the role of data-driven pedagogy. While their work provided foundational insights into data utilization for education, it lacked the integration of intelligent systems capable of providing interpretability. Later, Kelleher and Tierney (2018) highlighted the potential of data science for educators, proposing machine learning as a means to uncover hidden learning patterns; however, their study also did not address the critical issue of model transparency.

As digital learning platforms evolved, researchers such as Liang et al. (2019) investigated learner behaviors in massive open online courses (MOOCs) through machine learning clustering techniques. Their findings showed that learning data could be effectively used to categorize students based on engagement and performance metrics. Nevertheless, these models operated as black boxes, offering little interpretive value to instructors or learners. Recognizing this limitation, Xu and Lehman (2020) introduced Explainable AI (XAI) into educational data mining, demonstrating how model-agnostic explanation techniques like LIME could enhance transparency in predictive models. Their work marked a shift from pure prediction to interpretability, though it remained at the conceptual and prototype stage.

Building upon these developments, Khosravi et al. (2021) proposed a personalized learning framework using Explainable AI, enabling instructors to understand the rationale behind learning recommendations. Their model successfully linked explainability with adaptive learning pathways, yet it was not tailored for code-based or Online Judge environments. Around the same time, Santos and Rodrigues (2020) experimented with supervised learning algorithms to predict student academic success, but their approach lacked interpretive mechanisms, making it difficult for educators to trust or apply the outcomes effectively.

Recent research has increasingly turned toward AI in programming education. Zhou et al. (2022) developed intelligent assessment models using Online Judge data, analyzing coding submissions to classify student skill levels. Although their system effectively measured performance, it failed to provide reasoning behind classifications. Similarly, Rahman and Ahmed (2022) employed Random Forest and Decision Tree algorithms for modeling student programming strengths, but their work also lacked explainable outputs, leaving instructors uncertain about how features contributed to the final assessment. Zhang et al. (2021) proposed behavioral analytics for Online Judge platforms using time-based and syntax metrics; however, their approach relied on statistical analysis rather than AI explainability.

The growing interest in transparent educational AI systems was further highlighted by Choudhury and Singh (2021), who emphasized the importance of explainability in digital learning environments. Their XAI framework successfully identified learning difficulties, providing interpretable feedback to learners. However, their work primarily analyzed quiz interactions, not complex coding behaviors. Similarly, Kaur and Sharma (2023) combined SHAP and LIME with educational prediction models, making risk predictions interpretable and visually explainable, but their experiments were limited to grade prediction datasets.

In the programming education domain, Mishra and Pradhan (2023) used XGBoost to develop an AI-driven profiling system for online coding platforms. Their model demonstrated strong predictive performance but operated as a black box, lacking transparency in its results. Addressing this, Nguyen and Li (2023) experimented with deep learning models like CNN-LSTM to analyze sequential student submission data in Online Judge Systems, achieving high accuracy but minimal interpretability—reaffirming the need for explainable frameworks. Meanwhile, Reddy and Rao (2023) introduced feature importance analysis for classroom ML models, showcasing the importance of educational explainability, though their focus remained on academic grades rather than coding analytics.

The relevance of XAI in education was further strengthened by Wang and Tang (2024), who emphasized the necessity of building trustworthy AI systems in learning environments. They highlighted that explainability not only improves transparency but also fosters fairness and ethical accountability in AI-driven educational assessments. Sarkar and Mitra (2024) extended this notion to coding education by applying SHAP-based feature attribution methods to interpret student programming behaviors, though their work was constrained by a small dataset. Similarly, Ghosh and Banerjee (2024) combined interpretable machine learning with adaptive learning analytics, offering real-time, personalized insights to learners. Their findings confirmed that integrating interpretability enhances the overall learning experience and builds user trust.

From the reviewed literature, it is evident that while significant progress has been made in using machine learning and AI for education, most systems prioritize accuracy over interpretability. The majority of existing studies either lack explainability or fail to apply XAI in the context of Online Judge Systems, where coding submissions provide rich behavioral and temporal data. Although a few studies have successfully integrated explainability techniques like SHAP and LIME, they are primarily confined to traditional academic prediction models rather than code-based student profiling. This gap highlights the pressing need for a unified, transparent framework that not only predicts but also explains student learning behaviors and programming patterns.

Thus, the current research—"Decoding Learners: Explainable AI for Intelligent Student Profiling in Online Coding Platforms"—builds upon these prior works by integrating Explainable AI techniques directly into Online Judge Systems. The proposed system focuses on interpreting coding behavior, identifying skill progression, and visualizing reasoning behind each learner's profile, thereby bridging the gap between prediction accuracy and human interpretability in educational AI.

3. SYSTEM ANALYSIS AND DESIGN

EXISTING SYSTEM

The 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 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.

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.

PROPOSED SYSTEM

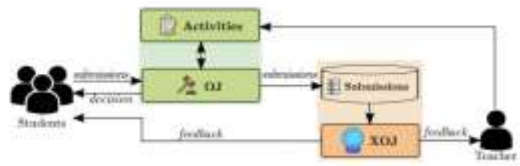
Considering all the above, this work presents a method to identify student profiles in educational OJ systems with the aim of providing feedback to both the students and the instructors about the development of the task. More precisely, the proposal exclusively relies on the meta-information extracted 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 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.

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.

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 Online Student's Profile judgement Ratio, Download Predicted Data Sets, View Online Student's Profile judgement Type Ratio Results, View All Remote Users.

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT STUDENT'S PROFILE DETECTION TYPE, VIEW YOUR PROFILE.

ALGORITHIM

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

Logistic regression Classifiers

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

6. SCREEN SHOTS

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

7. CONCLUSION

The proposed "Decoding Learners" framework bridges the gap between performance analytics and educational interpretability in Online Judge Systems. By incorporating Explainable AI, it empowers both instructors and students to understand not just what the results are, but why they occur. This transparency fosters trust, supports personalized education, and promotes data-driven academic development. Future work will explore integrating reinforcement learning and dynamic dashboards for real-time visualization of learning progression.

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 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.