# Identifying Student Profiles Within Online Judge Systems Using Explainable Artificial Intelligence

**Mohammed Munawar[1], Mohammed Mustafa[2], Mohammed Subhan Uddin[3], Mr. Yellaiah Ponnam[4]**
[1,2,3]B.E. Student, Department of IT, Lords Institute of Engineering and Technology, Hyderabad.
[4] Assistant Professor, Department of IT, Lords Institute of Engineering and Technology, Hyderabad.
yellaiahp@lords.ac.in

***Abstract— Online Judge (OJ) systems are widely used in programming courses to provide fast and objective evaluation of students' code. However, these systems usually deliver only a binary outcome—pass or fail—which offers limited educational value. To address this limitation, we propose a learning-based approach that leverages the behavioural data captured by OJ systems to generate richer and more informative feedback. Our method employs Multi-Instance Learning and traditional Machine Learning techniques to model student behaviour, while Explainable Artificial Intelligence (XAI) ensures that predictions and feedback remain interpretable and actionable. The approach was validated on a case study involving 2,500 submissions from 90 students in a Computer Science programming course. Results show that the model can accurately predict student outcomes based solely on behavioural patterns and identify at-risk groups. This contributes valuable insights for both learners and instructors, enhancing guidance, early intervention, and teaching strategies beyond binary evaluation.***

***Keywords: Predictive models, Machine learning, Task analysis, programming profession.***

## I. 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 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]. 2 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 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 characterizes 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 3 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 metainformation 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 metainformation of the OJ.

## II. RELATED WORK

### *A. Existing research and solution*

The integration of Online Judge (OJ) systems in computer science education has created new opportunities for tracking and analysing students' learning behaviors through their coding activities. Platforms such as Codeforces, LeetCode, and HackerRank capture detailed submission logs, including success rates, time spent on tasks, and error patterns. This data has been leveraged in numerous studies to identify and classify students based on their programming habits and learning performance.

Initial research efforts in this domain primarily relied on descriptive analytics and statistical methods to interpret student behaviour. For example, researchers examined patterns like the number of attempts per problem, response time, and frequency of engagement to evaluate students' problem-solving strategies. Such work helped distinguish learners by skill level and engagement intensity but lacked predictive power and adaptability to individual learning paths.

With the evolution of machine learning techniques, more advanced models have been introduced to uncover hidden patterns and automatically cluster students into distinct profiles. Studies have applied algorithms such as Artificial Neural Networks (RNNs) to model learning progression over time, and unsupervised learning methods like K-means clustering to group students based on behavioral similarities. These models demonstrated a strong ability to predict performance and categorize learners; however, their black- box nature often made it difficult to understand the reasoning behind the model decisions, raising concerns around transparency and trust.

To address this, recent work has shifted towards Explainable Artificial Intelligence (XAI), which aims to make machine learning models more transparent and interpretable. Techniques such as LIME (Local Interpretable Model- agnostic Explanations) and SHAP (SHapley Additive exPlanations) have gained popularity in educational contexts. These methods provide insights into which features most influence model predictions, enabling educators to understand and trust the outputs. Research has shown that interpretable models can enhance the decision-making process for instructors by highlighting factors such as student persistence, code complexity, and engagement frequency.

In the context of Online Judge systems, the application of XAI is still developing but has shown encouraging results. Some studies have utilized decision trees combined with SHAP values to interpret student performance data and highlight critical behavioral features. Others have proposed explainable pipelines that use interpretable machine learning models, such as Random Forests with feature importance analysis, to categorize learners and provide visual explanations for educators. These approaches not only improve the accuracy of student profiling but also offer actionable insights that can support personalized feedback, early intervention, and curriculum design.

### *B Problem Statement*

Online Judge (OJ) systems are widely used platforms in programming education, providing students with automated feedback on coding exercises and assessments. While these systems offer rich data on student behavior and performance, extracting meaningful insights from this data remains a challenge. Traditional analytical approaches often rely on statistical summaries or rule-based systems, which may overlook complex patterns in student activity and do not always provide interpretable results. As a result, educators may struggle to personalize instruction or intervene effectively to support at-risk students.

Recent research has attempted to model student behavior using machine learning techniques to categorize learners based on problem-solving strategies, submission patterns, and success rates. However, many of these models act as “black boxes,” offering limited transparency into how classifications are made. This lack of explainability reduces trust in the models and hinders their integration into educational practices, where understanding the why behind a prediction is often as important as the prediction itself.

To address these limitations, researchers have begun exploring the use of Explainable Artificial Intelligence (XAI) to identify student profiles within OJ systems. XAI methods aim to balance model accuracy with interpretability, enabling educators to not only detect learning patterns but also understand the reasoning behind model decisions. This approach holds significant potential for enhancing student support, optimizing curriculum design, and fostering adaptive learning environments.

Despite its promise, the application of XAI in the context of OJ systems is still in its infancy. There is a need for systematic investigations that combine behavioral data from OJ platforms with interpretable machine learning techniques to derive actionable student profiles. These profiles can provide insights into problem-solving habits, perseverance, and learning styles, ultimately contributing to more effective educational intervention

## III RESEARCH METODOGLY

This research presents a methodology for identifying an individual's learning style through on-the-job (OJ) learning techniques while evaluating all pertinent characteristics. This approach facilitates feedback for both educators and learners. The proposal emphasizes the integration of explainable artificial intelligence (XAI) methods to enhance the understanding of expected behaviors, alongside a multiple-instance learning (MIL) framework that enables the system to autonomously identify these profiles. The meta-data produced by these OJ systems is crucial for accomplishing this objective. A new policy is introduced to convert the MIL representation into a machine learning (ML) representation suitable for this task, thereby allowing the application of XAI to the MIL challenge. A three-year case study involving over 2,500 submissions from two distinct projects in a programming course was conducted to evaluate the proposed methodology. To determine the effectiveness of the concept, a review and comparison of more than twenty learning-based strategies utilizing ML, MIL, and MIL to ML mapping techniques were performed. Based solely on the meta-information derived from the OJ, the results indicate that the proposal accurately models the student user profile and provides a highly precise prediction of the students' likelihood of passing or failing the specific assignment. Transparency strategies, for instance, are methods that facilitate a clear understanding of the model's functioning. Post-hoc explanations are theoretical endeavors aimed at elucidating the model's development process. In contrast to transparency-focused strategies, this study prioritizes the latter scenario, thereby negating the necessity for each learning-based model to be specifically tailored to the task at hand[19].

The proposed method for quantitatively addressing these questions, which includes the following steps, is depicted graphically in Figure 1.

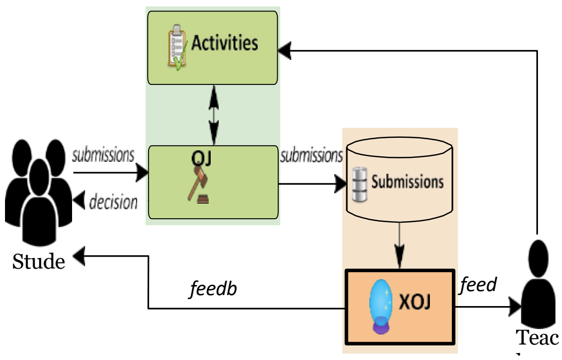


Figure.1. Proposed Methodology Block diagram

1) The instructor characterizes the various tasks to be settled by the understudies and designs the OJ framework as needs be.
2) The students respond to the given task and present their solutions.
3) The OJ assesses these entries and gives the understudies a remedy mark solely founded on the assessment of the submitted programming codes.
4) Concurrently, these entries are handled by an extra module XOJ in the plan that gives criticism to both the educator who might adjust the difficulty of the undertaking and the understudies who might accordingly change their obligation to the assignment. Note that this component addresses the center component of the work as it is intended to display the client conduct thinking about a regulated learning system.

This proposed system grades the assignments automatically and very quickly. There are few learning based algorithms to predict he student behavior and also evaluate performance of student. The algorithms are described below.

## IV ALGORITHMS

There are various machine learning algorithms to discusses the online judge system are listed below.

### A. Decision tree classifiers

Decision tree methods appear to be applicable in a variety of circumstances. The most essential aspect about them is that they can learn to use the information you offer to draw complex judgments. Training sets can be used to build decision trees. This demonstrates how to construct an object that resembles a collection of items (S) from the classes C1, C2 ,... Ck.

Step 1If every element in the S decision tree belongs to the same class, the leaf will be labeled with the class Ci.

Step 2. If T is not else given, define it as a test that returns on. The examination divides S into groups S1, S2,..., Sn, with each item in Si representing a single probable outcome in T. This is the case because each element in S has a unique possible outcome for T. The decision tree starts at point T, and the same procedures are used to build a child decision tree on the set Si for each outcome Oi.[19].

### B. Gradient boosting

Techniques in machine learning, such as gradient boosting, are employed to address various challenges, including both regression and classification tasks. A prediction model is generally formed by combining several weak predictive

ISSN 2277-2685
IJESR/April-May. 2025/ Vol-15/Issue-2s/1590-1597
**Mohammed Uzaafar Arfath *et. al.*, / International Journal of Engineering & Science Research**

models, such as decision trees. When a decision tree acts as the base learner, gradient-boosted trees come into play. In most cases, they demonstrate superior performance compared to random forests. Similar to other boosting methods, gradient-boosted trees are built incrementally. However, they surpass earlier techniques by enabling the optimization of any differentiable loss function.

**C. K-Nearest Neighbors (KNN)**

This straightforward yet highly effective categorization method classifies objects according to their similarities. It operates slowly and is non-parametric, meaning it does not "learn" until it encounters a test case. By utilizing the training data, we identify the K-nearest neighbors of the newly categorized data.

**D. Logistic regression Classifiers**

Logistic regression analysis evaluates a collection of categorical independent variables that contribute to the explanation of a categorical dependent variable. The dependent variable is binary, possessing only two possible outcomes: zero and one, or yes and no. The term "logistic regression" is widely recognized to describe this type of analysis. In cases where the dependent variable encompasses three or more categories, such as married, single, divorced, or widowed, multinomial logistic regression is typically utilized. Although multiple regression employs a different dataset to represent the dependent variable, the fundamental methodology remains consistent. Both discriminant analysis and logistic regression serve as effective techniques for distinguishing between categorical response categories. A significant number of statisticians assert that logistic regression generally surpasses discriminant analysis in modeling various scenarios. However, logistic regression encounters limitations when the independent variables do not adhere to a normal distribution, a situation that is not problematic for discriminant analysis.

**E. Naïve Bayes**

The naïve Bayes method is a supervised learning approach that posits that the presence or absence of a particular attribute within a class does not influence the presence or absence of other attributes. Despite this assumption, it has proven to be both effective and practical. In terms of functionality, various supervised learning methods are comparable. The literature provides numerous explanations for this. In this discussion, we will examine a scenario involving representation bias[19]

**F. SVM**

A discriminant machine learning methodology is employed to establish a discriminant function that effectively infers labels for new instances derived from an independent and identically distributed (iid) training dataset. This approach is utilized for job classification. In this classification process, a discriminant classification function is utilized to allocate a data point x to a designated class.

However, the implementation of generative machine learning techniques necessitates the prior construction of conditional probability distributions. When incorporating outlier detection into the prediction framework, discriminant methods typically require fewer training samples and computational resources compared to generative methods.

## V RESULT & DISCUSSION

When performing comparisons among various algorithms, they show various accuracy values with respect to the various algorithms.

These are listed in the following Table 1. RF scheme with MIL-to-ML mapping is the best performing strategy XAI scheme provides feedback and high recognition rate and Identifies prone-to-fail student groups and profiles, offering valuable feedback

**Table.1. Comparison of various algorithms with their accuracy**

| S.No | Name of the Algorithm | Accuracy% |
|---|---|---|
| 1 | Artificial Neural Network | 63.0 |
| 2 | Navie Bayes | 70.5 |
| 3 | SVM | 66.5 |
| 4 | Logistic Regression | 68.0 |
| 5 | Gradient Boosting Classifier | 67.5 |
| 6 | Decision Tree Classifier | 64.0 |
| 7 | KNN Classifier | 58.5 |

ISSN 2277-2685
IJESR/April-May. 2025/ Vol-15/Issue-2s/1590-1597
Mohammed Uzaafar Arfath *et. al.*, / International Journal of Engineering & Science Research

From figures 2 to 5 shows the resultant screenshots of Online judge system using XAI.

Figure 2: Service Provider Login Page

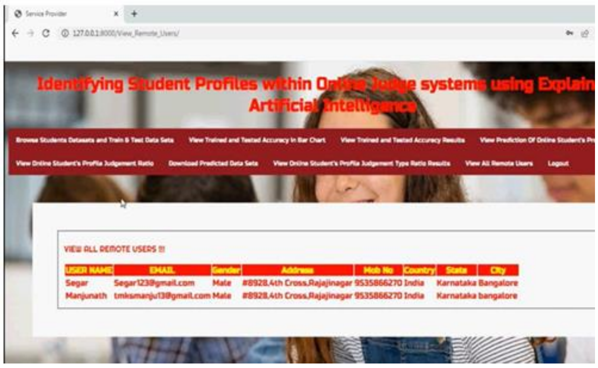

Figure 3: View Remote User Profile

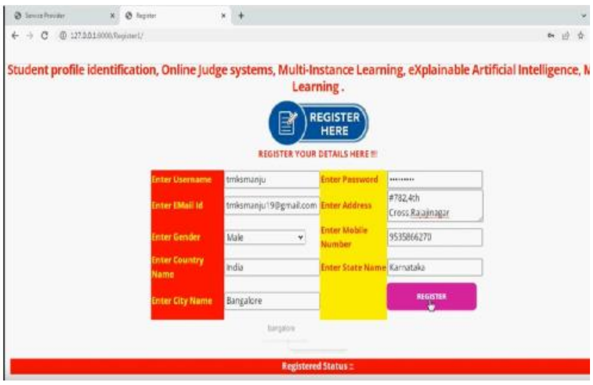


Figure 4: Remote User Register Page

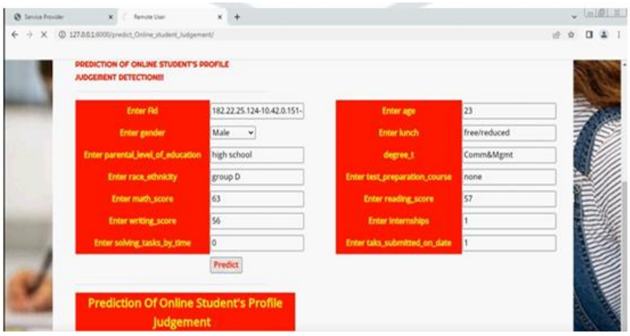


Figure 5: View Prediction Results

## VI CONCLUSION

This research presents a novel approach that integrates Explainable Artificial Intelligence (XAI) with Online Judge (OJ) systems to enhance the assessment and feedback mechanisms in programming education. By employing machine learning techniques, specifically Multi-Instance Learning (MIL) and decision tree models, the study successfully identifies distinct student behavior patterns based on their code submission data. The model demonstrates a significant capability to predict student outcomes—pass or fail—by analyzing these behavioral patterns.

A key contribution of this work is the incorporation of XAI, which provides interpretable insights into the decision- making process of the predictive models. This transparency allows educators to understand the underlying factors influencing student performance, facilitating the identification of at-risk students and enabling timely interventions. Moreover, students receive meaningful feedback that can guide their learning strategies and improve their problem-solving skills.

One of the most important contributions of this research is its ability to model and predict student outcomes based on their interaction with the OJ system, particularly through an analysis of their code submissions and solution behaviors. By applying machine learning algorithms such as Multi-Instance Learning (MIL) and decision tree models, the study identifies distinct behavioral profiles that categorize students according to their programming proficiency, problem-solving approach, and submission patterns. These profiles provide educators

with a nuanced understanding of students' strengths and weaknesses, enabling more precise identification of students who may require additional support or challenge.

A unique aspect of the research lies in its emphasis on the explainability of the AI models used. Unlike traditional "black-box" models, XAI offers insights into how the model arrives at its predictions, allowing for more trust in the AI's decision-making process. For educators, this transparency is invaluable, as it enables them to trace the factors that contribute to a student's performance prediction. With this knowledge, instructors can tailor interventions more effectively, providing personalized support for individual learners. Additionally, students can benefit from more constructive feedback, helping them to understand their errors, identify areas for improvement, and adopt strategies to enhance their coding and problem-solving skills

## VII.REFERENCE

[1] P. Naval and N. Pandey, ‘‘What makes a consumer redeem digital coupons? Behavioral

[7] customer loyalty segment-based customer churn prediction methodology,’’ Intell. Inf. Res., vol. 26, no. 4, pp. 111–126, 2020.

[8] R. M. Gubela, S. Lessmann, and S. Jaroszewicz, ‘‘Response transformation and profit decomposition for revenue uplift modeling,’’ Eur. J. Oper. Res., vol. 283, no. 2,
pp. 647–661, Jun. 2020.

[9] M.-S. Chang, H. Kim, and Joong, ‘‘A customer segmentation scheme base on big data in a bank,’’ J. Digit. Contents Soc., vol. 19, no. 1, pp. 85–91, 2018. N. Chang, ‘‘Improving the effectiveness of customer classification models: A pre-segmentation approach,’’ Inf. Syst. Rev., vol. 7, no. 2, pp. 23–40, 2005.

[10] C.-F. Tsai and Y.-H. Lu, ‘‘Customer churn prediction by hybrid neural networks,’’ Expert Syst. Appl., vol. 36, no. 10,
pp. 12547–12553, Dec. 2009.

[11] Y. Xie, X. Li, E. W. T. Ngai, and W. Ying, ‘‘Customer churn prediction using improved balanced random forests,’’ Expert Syst. Appl., vol. 36, no. 3, pp. 5445–5449, Apr. 2009.

[12] S.-Y. Hung, D. C. Yen, and H.-Y.Wang, ‘‘Applying data mining to telecom churn

insights from grounded theory approach,’’ J. Promotion Manage., vol. 28, no. 3, pp. 205–238, 2021.

[2] C. Hung and C. F. Tsai, ‘‘Market segmentation based on hierarchical selforganizing map for markets of multimedia on demand,’’ Expert Syst. With Appl., vol. 34, pp. 780–787, Jan. 2008.

[3] G. Nie, ‘‘Finding the hidden pattern of credit card holder’s churn: A case of China,’’ in Proc. Int.Conf. Comput. Sci. Cham, Switzerland: Springer, 2009, pp. 561–569.

[4] A. D. Athanassopoulos, ‘‘Customer satisfaction cues to support market segmentation and explain switching behavior,’’ J. Bus. Res., vol. 47, no. 3, pp. 191–207, Mar. 2000.

[5] C. Hung and C. F. Tsai, ‘‘Market segmentation based on hierarchical selforganizing map for markets of multimedia on demand,’’ Expert Syst. With Appl., vol. 34, pp. 780–787, Jan. 2008.

[6] H.-S. Kim and H. Seung-Woo, ‘‘A two-dimensional

management,’’ Expert Syst. Appl., vol. 31, no. 3, pp. 515–524, Oct. 2006.

[13] J. Wen and W. Zhou, ‘‘An improved itembased collaborative filtering algorithm based on clustering method,’’
J. Comput. Inf. Syst., vol. 8, no. 2,pp. 571–578, 2012.

M. Pham and Cuong, ‘‘A clustering approach for collaborative filtering recommendation using

social network analysis,’’ J. Univers. Comput. Sci., vol. 17,
pp. 583–604, Feb. 2011.

[14] W. Jo-Ting, L. Shih-Yen, andW. Hsin-Hung, ‘‘A review of the application of RFM model,’’ African J. Bus. Manage., vol. 4, no. 19, pp. 4199–4206, 2010.

[15] J. T. Wei, S.-Y. Lin, Y.-Z. Yang, and H.-H. Wu, ‘‘The
application of data mining and RFM model in market segmentation of a veterinary hospital,’’
J. Statist. Manage. Syst., vol. 22, no. 6, pp. 1049–1065, Aug. 2019.

[16] M. Pakyurek, M. S. Sezgin, S. Kestepe, B. Bora, R. Duzagac, and O. T. Yildiz, ‘‘Customer

clustering using RFM analysis,'' in Proc. 26th Signal Process. Commun. Appl. Conf. (SIU), May 2018, p. 2.

[17] P. A. Sarvari, A. Ustundag, and H. Takci, ''Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis,'' Kybernetes, vol. 45, no. 7, pp. 1129–1157, Aug. 2016.

[18] F. Tian, ''Learning deep representations for graph clustering,'' in Proc. AAAI Conf. Artif. Intell., 2014, pp. 1293–1299

[19] A. Pasumarthi and U. Darapu, "Explainable AI: Providing Student Feedback via Online Judge," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 8, Aug. 2024.