

Predicting Student Performance in Software Engineering Education Using Random Forest: A Data-Driven Approach Based on Subjective Assessments

Linling Zhu*

School of Information Science and
Technology
Sanda University
Shanghai, China
zlling@sandau.edu.cn

Siyu Zhang

School of Information Science and
Technology
Sanda University
Shanghai, China
zhangsiyu0066@gmail.com

Yi Wei

School of Information Science and
Technology
Sanda University
Shanghai, China
weiyi2003@foxmail.com

Xiaoyue Tu

School of Information Science and
Technology
Sanda University
Shanghai, China
sophie.tuxy@icloud.com

Yuxiang Huang

School of Information Science and
Technology
Sanda University
Shanghai, China
YuxiangHuang6@outlook.com

Meng Wu

School of Information Science and
Technology
Sanda University
Shanghai, China
mwu@sandau.edu.cn

Abstract

The research aims to study the efficacy of random forest algorithm modelling student grades in software engineering education and compares the predictions made by the model with actual student grades. The data are collected from 88 students from Sanda University in the course "Software Testing Practice" including both subjective ratings by teachers and team leaders across five dimensions: testing requirements, testing plans, testing cases, defect discovery, and testing reports; and students' actual grades in the course "Software Engineering and Project Management" which are objective assessment criteria. It will be compared between two models, Random Forest versus AdaBoost, to determine which is better predicting the students' grades. The overall model designed in this study with subjective ratings from a single course to predict future grades in related courses can be demonstrated to show different performance using their respective different performance models. It has been shown through correlation studies that the random forest model exhibits a greater correlation with actual scores (0.629) than does the AdaBoost model (0.475). More evaluation with measurements such as MAE, MPE, MAPE, as well as R^2 confirmed the advantages of the random forest model in terms of accuracy and reliability. The model can be further enhanced and made adaptable to various learning environments to more accurately reflect and inform better decision-making in the future and ongoing software engineering education improvements.

*Corresponding author



This work is licensed under a Creative Commons Attribution International 4.0 License.

DEIT 2025, Nanjing, China

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1324-8/2025/02

<https://doi.org/10.1145/3732299.3732326>

CCS Concepts

• **Computing methodologies** → Machine learning; Machine learning approaches; Instance-based learning.

Keywords

Software Testing Education, Random Forest, Machine Learning, Project Driven Learning, Correlation Coefficient

ACM Reference Format:

Linling Zhu, Siyu Zhang, Yi Wei, Xiaoyue Tu, Yuxiang Huang, and Meng Wu. 2025. Predicting Student Performance in Software Engineering Education Using Random Forest: A Data-Driven Approach Based on Subjective Assessments. In *2025 International Conference on Digital Education and Information Technology (DEIT 2025)*, February 21–23, 2025, Nanjing, China. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3732299.3732326>

1 Introduction

1.1 Research Background

See has been on the platform for more than fifty years; its prime mission is to equip students with solid fundamentals to enable them to thrive in the fast-changing technology world [1]. Software engineering education aims to fuse theory and practice so that students would strongly know core concepts and principles and apply them efficiently in solving real-world problems. The aim is thus not only to impart knowledge but also to train professional personnel who would easily adapt to the changing needs in the industry [2].

The software industry is witnessing an increasing convergence of Artificial Intelligence (AI) and Machine Learning (ML) technologies as the impetus for intelligent advances and increased efficiency [3]. These technologies have, thus far, seen adoption across a wide variety of industries as organizations strive to apply AI and ML technologies to their products and services for a competitive edge in the marketplace [4]. Consequently, it becomes imperative that AI and ML become part of software engineering education. This

way, the students will develop capabilities of applying these advanced technologies in solving complex problems while mastering traditional theories to address the changing needs of the software industry.

1.2 Research Questions

This research studies the relationship between subjective and objective grades in software engineering education. The specific research questions are as under:

RQ1: To what extent the random forest algorithm is able to provide predictions of future student performances, using subjective scores, such as: ordinary grade and assessments of individual processes (test requirements, test plans, test cases, defect findings, and test reports) carried out by lecturers and team leaders?

RQ2: Which model, the Random Forest model or the AdaBoost model, gives better prediction of students' grades and correlates more with objective scores?

2 LITERATURE REVIEW

2.1 Software Testing Teaching Methods

In recent years, it has been noted that software testing education has been gradually moving towards real life and production methods of practice of all activities, encouraging the practical application of students' skills and teamwork development. Most literature studies have mainly included the agile teaching method, morph-and-test (MT) teaching method, and gamification teaching method.

Active teaching methods used include gamification, dojo, and fun teaching, whose intention is to engage the students in their learning journey [5]. In addition, the curriculum has incorporated agile practices to enable students to learn the underlying principles of software testing, development methodologies, and quality strategies to supplement these with [6].

To Liu et al., a lean, automated and cost-effective methodology for testing has been identified that suits end-user programmers at the classroom level. Using a combination of lecture and tutorial, students could comprehend as well as practice the basics and automated realizing deformation testing [7].

Designing a game called 'CODE DEFENDERS', Clegg et al. used "in a gamified mode" the concepts of testing, such as variant testing, to teach software testing. Gradually, students mastered basic testing skills, such as statement and branch coverage, in that game. The results also demonstrated that gamification could significantly improve the interest in learning and comprehension and application abilities related to testing concepts among students [8].

People, at present, believe that current teaching systems in software testing are leaning more toward practical interaction and are making amalgamation of agile methods with morphing testing as well as gamified teaching-another step forward in enhancing interest and participation in testing besides augmenting practical skills.

2.2 Random Forest Algorithm

In the recent years, statistical methods and machine-learning algorithms have found their way into many areas of social sciences. Liu et al. showed that random forest regression outperforms ordinary least squares regression in predicting recession in terms of

goodness of fit [9]; meanwhile, Basuchoudhary et al. describe the application of different statistical learning algorithms to predict economic growth and recession in their work [10]. For environmental sciences, Alshannaq et al. demonstrated that random forest possesses the best predictive performance with respect to dealing with historical data and considered multiple predictor variables [11].

In software engineering, machine learning, in general, shows much promise, and random forest algorithms, in particular, have demonstrated potential. Petkovic et al. have successfully improved through the application of more than 100 objective and quantitative measures of teamwork activities on the prediction of student learning outcomes to over 70% accuracy [12]. This clearly demonstrates the potential of machine learning in education, especially teamwork assessment, and shows its advantages for predicting student performance.

Li et al., on the other hand, introduce a completely novel approach to assessing students' learning methods that break away from the barrier of assessing single-influencing factors to a comprehensive assessment of the correlation between teacher assessment and student self-assessment for a more scientific and rational standard of assessment [13]. This assessment method is applicable in a wide range of scenarios. It can be operated easily and integrate course learning contents with personalized teaching plans, thus strongly supporting the improvement of teaching quality.

To summarize, the use of machine learning, especially random forest algorithms, in social sciences, environmental sciences, and education proved its high power in prediction. These algorithms are mechanisms to further improve prediction accuracy through the combination of multiple evaluation factors, thus providing strong data support to enhance the improvement of teaching quality.

2.3 AdaBoost Algorithm

Computer technology is evolving, so software engineering programs can foster problem-solving skills needed by students. Firdaus Zainal Abidin et al. used a combination of the AdaBoost algorithm and a multilayer perceptron (MLP) model to improve the accuracy of predicting student performance. Their model presented a great increase in prediction and generalization. The predictive model allows the teaching practitioners to evaluate student learning outcomes with more accuracy so they can teach in a more personalized manner [14].

To enhance the performance of the predictive model, Wang et al. examined the application of AdaBoost regression (ABR) to increase the predicting capabilities of Infrared Index (IRI). They compared the ABR models to linear regression (LR) models within a multi-element pseudo-dyadic graph generator (MEPDG). Their results emphasize that ABR outperforms classical regression approaches in higher prediction accuracy [15].

In addition, Praveena et al. studied the AdaBoost algorithm, integrating several machine learning models with an optimization approach. Their observation corroborated that the AdaBoost is effective in gaining prediction abilities and has wide applications influencing its might towards a number of prediction tasks [16].

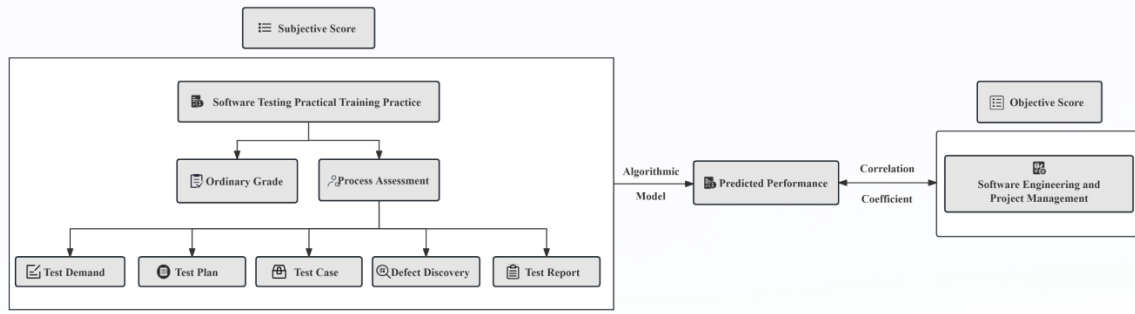


Figure 1: Research Framework.

This summarizes the great potency of AdaBoost and its derivatives for undertaking educational solutions observed in these studies. By providing sound scientific bases for predicting student performance, these models support the improvement of academic assessment and instruction. Furthermore, not restricting the adaptability of AdaBoost to educational arenas, it offers unparalleled perspectives and techniques for predictive analysis across a spectrum of interdisciplinary fields. All these advancements further reinforce its applicability in academic studies and the real world.

3 RESEARCH DESIGN

3.1 Research Approach

The purpose of the study is to investigate the success of applying a predictive model that mingles subjective assessment scores of one course with the forecasting of objective performance in subsequent Software Testing courses. The dataset consists of 88 students from the Class of 2020 in the Software Engineering Department of Sanda University.

The subjective scores constitute those granted in the course of Software Testing Practical Training Practice and included ordinary grades and process assessment. The process assessment comprised the following aspects: 1. Test Demand: the degree of being able to accurately identify and formally document test requirements; 2. Test Plan: the degree of being able to formally document a complete test strategy; 3. Test Case: the degree of being able to create effective and efficient test cases; 4. Defect Discovery: the degree of being able to find and document defects in the testing; 5. Test Report: the degree of being able to detail and organize test reports. The above characteristics were meant to give an overview of the student measurements in the testing process concerning level of competence and practice.

The objective grades represent the test marks of the student on the following semester in the course “Software Engineering and Project Management.” A Random Forest-based predictive model was used to investigate subjective-objective assessment relationships and the scope of predicting future performance with past course grades. The model output, together with correlation analysis to determine the strength and significance of the relationship between subjective assessments and objective performance, provided valuable pointers for improving methods of student assessment and academic intervention in software testing education (see Figure 1).

3.2 Data Processing

Handling missing values in the data pre-processing stage assures accurate and convenient analysis of the data from which the analysis can be performed. Missing data usually seriously affects outputs, and so it is important to treat missing values properly. In this research, missing data were handled by mean estimation, i.e., for each missing value in a column of the data set, the mean of that specific column is calculated to fill in the blank. This method is useful not only for minimizing losses but also for preserving data integrity and ensuring the validity and reliability of future analyses.

It is now important for discussion of how to treat missing data to also look at the empirical nature of the missing data in a particular dataset and its relevance for model performance. This study looks at the kind, volume, and distribution of missing data that, in turn, help to determine their best management. Proper analysis of missing data pattern leads to its classification as either being missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR), which has great bearing on the interpolation strategy chosen. In fact, mean estimation is used in this study because it is easy and effective, but other approaches, such as single-interpolation (like median or multinomial interpolation) and multiple interpolations (like predictive mean matching, regression-based interpolation), can be examined as well. These strategies bear different downside impacts on model performance in terms of retaining data variance and minimizing bias. Towards this effect, further research should compare these methodologies currently so that the effect may later be assessed on predictive accuracy, generality of model, and, subsequently, select the best one in terms of impact as kneaded through empirical evidence.

Correlation with both subjective and objective ratings was the main focus of this study after data pre-processing and variance reduction. A coefficient-correlation matrix analysis was applied to address this study’s specification of a linear correlation between research variables. The correlation coefficient ranges between -1 and 1 and is a measure of both strength and direction of a bilateral linear relationship between two variables. A correlation coefficient equal to 1 means that a perfect positive correlation exists between the two variables, one equal to -1 means perfect negative correlation, and equal to null means no linear correlation is observed. In so doing, a correlation coefficient was calculated to be able to find

quantitative relationships among the various scoring factors, thus establishing the basis for further modeling.

Following preprocessing of the data, the focus of this research was on the correlation of the variables in the dataset: namely, on the correlation between subjective and objective ratings. To analyze the linear relationship present among the variables, a coefficient-correlation matrix analysis was used. The correlation coefficient has a range between 1 and -1 and is a measure of both strength and direction of a bilateral linear relationship between the two variables. A correlation coefficient equal to one indicates that a perfect positive correlation exists between the two variables, one equal to negative one indicates perfect negative correlation, while equal to null means no linear correlation is observed. This work used calculation of the correlation coefficient to quantitatively analyze the relationships among different scoring factors, thus providing a solid basis for further modeling.

After data pre-processing this study focused on the correlation between the variables in the data set, especially with respect to the correlation between subjective and objective ratings. This study carried out a correlation coefficient matrix analysis in order to determine the linear relationship between the variables. The correlation coefficient is a measure defined between -1 and 1 and is used to determine the strength and direction of a linear relationship between two variables. A correlation coefficient of +1 implies perfect positive correlation, -1 implies perfect negative correlation, and 0 indicates non-linearity. A correlation coefficient was calculated to find out the quantitative relationships among various scoring factors and thus develop a firm basis for further modeling.

For example, a correlation analysis is said to be statistically significant if the absolute value of the correlation coefficient is greater than 0.4; if the absolute value is less than that, the correlation is deemed of weak strength and said to lack statistical significance. This threshold prescribes a boundary for weak correlations that could distort the analysis and guarantee the medicine's brilliance.

In addition to the previous analysis, heat maps were employed for further validating correlations (see Figure 2). The strength of correlations is thus graphically depicted in heat maps, aiding in the identification of significant relationships between variables. Such an approach not only assists in understanding the key relationships from the dataset but also informs subsequent modeling and decision-making.

This research studied the quality and reliability of the data through using a variety of techniques in data processing, such as treating missing values and performing correlation analyses for visual validation. In addition, such techniques offer the basis for exploring possible patterns in data, direct model construction, and optimize the decision-making process. This way, sound foundation has been laid towards academic research and educational improvement.

4 RESuLTS AND DISCUSSION

4.1 Performance Analysis of Random Forest Model

There is high agreement between the random forest model (orange dashed line and crosses) and actual values (blue solid line and dots) in terms of capturing the trend of changes in data (see Figure 3).

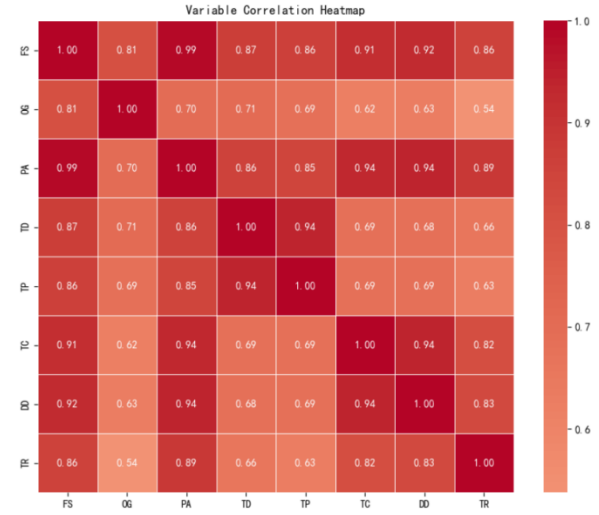


Figure 2: Correlation Matrix of Subjective Score. (FS: Final Score, OG: Ordinary Grade, PA: Process Assessment, TD: Test Demand, TP: Test Plan, TC: Test Case, DD: Defect Discovery, TR: Test Report)

For both models, deviations from the actual values occurred when close to sample index positions of around ten and fifteen. But for the random forest model, the deviation was small. Besides, the residual plot on the right confirms this point; the residuals (blue dots) of the random forest model are close to the zero line, which is also an indication of minuscule deviation between predicted and actual values. Particularly in low and high actual value ranges, the residual is small in the random forest model, meaning the prediction is accurate within these ranges. Hence, from the chart analysis, it could be concluded that the random forest model is better in predictive performance for this dataset rather than AdaBoost.

To evaluate the performance of the random forest algorithm and the AdaBoost algorithm, four main metrics were considered for the study: mean absolute error, mean percentage error, mean absolute percentage error, and R-squared.

The MAE score of the Random Forest model is 1.15, according to Table 1. Comparing this value with the one derived from the AdaBoost model, which is 1.98, indicates the superiority of the former model in terms of average error. Thus, it can be inferred that the Random Forest model has found more accurate predictions for student grades. In MPE, the value of Random Forest is placed at -1.05 and that of AdaBoost is at -0.10. The values are, however, negative and indicate a certain lean towards underestimating while at the same time exhibiting a huge MPE in Random Forest, suggesting that its predicted value is more deeply underestimated in comparison to the actual. Nonetheless, the variance of the MPE is relatively small thus does not affect the overall assessment of model accuracy in a highly significant way. MAPE is a measure that indicates the magnitude of prediction error relative to the actual value. The MAPE value of the Random Forest model is found to be 0.02 which is really less as compared to the MAPE value of the model AdaBoost which is 2.55, which goes on to prove that the Random Forest model

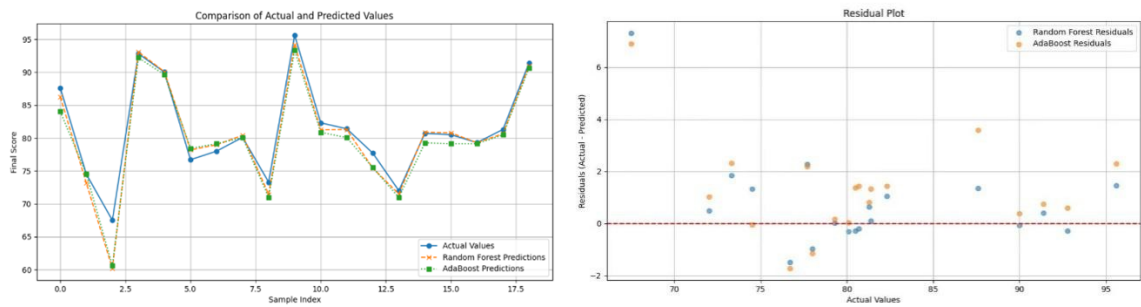


Figure 3: The Left Figure Is Random Forest Predictions and AdaBoost Predictions, The Right Figure Is a Residual Plot for Random Forest and AdaBoost.

Table 1: Random Forest Performance Evaluation.

Evaluation Indicators	Random Forest	AdaBoost
MAE	1.15	1.98
MPE	-1.05	-0.10
MAPE	0.02	2.55
R ²	0.93	0.87

Table 2: Correlation Index.

	Predicting Grades_Random Forest	Predicting Grades_AdaBoost	Objective Results
Predicting Grades_Random Forest	1	0.981	0.629
Predicting Grades_AdaBoost	0.981	1	0.475
Objective Results	0.629	0.475	1

is a winner with regards to prediction under accuracy terms. Thus, it can be said that the predicted value and the actual value have a rather small deviation which stands for high reliability. Also, R^2 is a crucial measure of fit for a model. It is the measure of the degree to which the explanatory variables in the model explain the changes in the target variables. The R^2 value of the Random Forest model is 0.93, whereas it is only 0.87 in the case of AdaBoost. The Random Forest model, therefore, explains approximately 93% of the variation in the performance, while AdaBoost explains only 87%. Such results show that the Random Forest model has a better capability to signify the key points important to the student’s academic performance, better fit, and a more realistic representation of the distribution of students’ performances.

Therefore, the random forest model would predict performance better than other algorithms to understand the high predictive precision and accuracy in forecasting students’ performance. In this sense, it explains and captures much better the important trends of academic performance among students and consequently serves with much reliability and accuracy as a predictive tool for educational evaluation and intervention. The findings strongly support the random forest model’s use in student achievement prediction fields and promise broad applications in related domains.

4.2 Correlation Analysis

According to Table 2, the Random Forest model achieves high predictive accuracy regarding the prediction of students’ scores. The correlation coefficient between the predicted and actual scores of the model stands at 0.629, which is significantly higher than the corresponding value obtained by the AdaBoost model at 0.475. The correlation coefficient is a statistical measure that indicates the degree of linear association between two variables, with a value of 1 being an indication of the strongest association. A high Random Forest correlation coefficient means it accurately simulates actual learning situations for students over knowledge acquisition, learning process activities, and final examination performance. Because of this high correlation, it means the Random Forest model tends to identify these key factors affecting students’ performance and provide even more reliable assessments of the students’ performance.

On the other hand, although AdaBoost shows some amount of predictive ability, low correlation coefficients between predicted and actual grades might suggest that the model is somewhat missing out on complex factors affecting students’ grades. The AdaBoost model could also be weaker than the Random Forest model in handling non-linear relationships or interactions among the features, which could take a toll on the model’s precision in predicting

student performance. Also, lower correlation coefficients suggest some issues with generalizability by the AdaBoost models, which means those might not be able to sustain that level of predictive performance on new data.

In sum, when it comes to selecting a model for predicting student performance, in this context, the Random Forest model gives a better option with stronger predictive ability and reliability. This allows one to measure students' academic performance and enables educators to better understand the learning needs of students in developing teaching strategies.

5 CONCLUSION

Results also show that while the Random Forest model is highly accurate and generalizable for predicting grades of students' academic performance, the AdaBoost model was not. Very significantly and positively correlated were subjective grades from the Software Testing Practice course and objective grades from the Software Engineering and Project Management course the following semester. Furthermore, the model was verified for accuracy and reliability through MAE, MPE, MAPE, and R^2 indices that further established the fit of the Random Forest model with a small prediction error.

The prediction model drawn up in this study has not only been found satisfactory in predicting students' future performance but is also an avenue for improving the educational evaluation system, thereby minimizing failure rates and allowing for targeted intervention in teaching. By utilizing the subjective scoring of one particular course to predict objective scores in later software testing courses, the prediction system would provide a way to early identify students whose performance is under threat in a structured manner. Future work will allow for model refinements that enhance its applicability across disparate educational environments, thereby ultimately improving decision support for further software testing education development.

Acknowledgments

This work was supported in part by the Sanda University Virtual Simulation Experiment Course under Grant A020201.22.905, in part by the Shanghai Municipal Key Course Project in Higher Education (AI+ Course) under Grant A020201.24.608.

References

- [1] Cico, O., Jaccheri, L., Nguyen-Duc, A., & Zhang, H. (2021). Exploring the intersection between software industry and Software Engineering education—A systematic mapping of Software Engineering Trends. *Journal of Systems and Software*, 172, 110736. <https://doi.org/10.1016/j.jss.2020.110736>.
- [2] Ouhbi, S., & Pombo, N. (2020). Software Engineering Education: Challenges and Perspectives. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 202–209. <https://doi.org/10.1109/EDUCON45650.2020.9125353>.
- [3] Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 291–300. <https://doi.org/10.1109/ICSE-SEIP.2019.00042>.
- [4] Nascimento, E., Nguyen-Duc, A., Sundbø, I., & Conte, T. (2020). Software engineering for artificial intelligence and machine learning software: A systematic literature review. *arXiv preprint arXiv:2011.03751*.
- [5] Elgrably, I. S., & Ronaldo Bezerra Oliveira, S. (2020). Model for teaching and training software testing in an agile context. *2020 IEEE Frontiers in Education Conference (FIE)*, 1–9. <https://doi.org/10.1109/FIE44824.2020.9274117>.
- [6] Elgrably, I. S., & Ronaldo Bezerra Oliveira, S. (2020). Construction of a syllabus adhering to the teaching of software testing using agile practices. *2020 IEEE Frontiers in Education Conference (FIE)*, 1–9. <https://doi.org/10.1109/FIE44824.2020.9274266>.
- [7] Liu, H., Kuo, F.-C., & Chen, T. Y. (2010). Teaching an End-User Testing Methodology. *2010 23rd IEEE Conference on Software Engineering Education and Training*, 81–88. <https://doi.org/10.1109/CSEET.2010.28>.
- [8] Clegg, B. S., Rojas, J. M., & Fraser, G. (2017). Teaching Software Testing Concepts Using a Mutation Testing Game. *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, 33–36. <https://doi.org/10.1109/ICSE-SEET.2017.1>.
- [9] Liu, S., Wang, X., Liu, M., & Zhu, J. (2017). Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1), 48–56.
- [10] Basuchoudhary, A., Bang, J. T., & Sen, T. (2017). Machine-learning techniques in economics: new tools for predicting economic growth. Springer.
- [11] Alshannaq, M., Imam, R., & Alsarayreh, D. W. (2023). Modelling of traffic accident severity in Jordan using machine learning. *International review of civil engineering*, 14(5).
- [12] Petkovic, D., Barlaskar, S. H., Yang, J., & Todtenhoefer, R. (2018, October). From explaining how random forest classifier predicts learning of software engineering teamwork to guidance for educators. In *2018 IEEE frontiers in education conference (FIE)* (pp. 1–7). IEEE.
- [13] Li, R. (2024). Evaluation of teaching effect of online education courses based on random forest. *Procedia Computer Science*, 243, 1059–1068.
- [14] Firdaus Zainal Abidin, A., Darmawan, M. F., Osman, M. Z., Anwar, S., Kasim, S., Yunianta, A., & Sutikno, T. (2019). Adaboost-multilayer perceptron to predict the student's performance in software engineering. *Bulletin of Electrical Engineering and Informatics*, 8(4), 1556–1562. <https://doi.org/10.11591/eei.v8i4.1432>.
- [15] Wang, C., Xu, S., & Yang, J. (2021). Adaboost Algorithm in Artificial Intelligence for Optimizing the IRI Prediction Accuracy of Asphalt Concrete Pavement. *Sensors*, 21(17), 5682. <https://doi.org/10.3390/s21175682>.
- [16] Praveena, M., & Jaiganesh, V. (2017). A Literature Review on Supervised Machine Learning Algorithms and Boosting Process. *International Journal of Computer Applications*, 169(8), 32–35. <https://doi.org/10.5120/ijca2017914816>.