JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Omar, M., Gilal, R., Md Rejab, M., Gilal, A. R., Almogahed A., & Nurhanifah (2025). Predictive modeling of software team performance based on gender and task complexity using logistic regression and decision tree. *Journal of Information and Communication Technology*, *24*(2),45-65. https://doi.org/10.32890/jict2025.24.2.3

Predictive Modeling of Software Team Performance based on Gender and Task Complexity using Logistic Regression and Decision Tree

[1]Mazni Omar, [2]Ruqaya Gilal, [3]Mawarny Md Rejab, [4]Abdul Rehman Gilal,
[5]Abdullah Almogahed & [6]Nurhanifah
[1,2&3]School of Computing, Universiti Utara Malaysia, Malaysia
[4]Knight Foundation School of Computing and Information Sciences,
Florida International University, USA
[5]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Malaysia
[6]School of Computing, Universitas Islam Negeri Sumatera Utara, Indonesia

*[1]mazni@uum.edu.my
[2]ruqayagilal1@gmail.com
[3]mawarny@uum.edu.my
[4]arehman@fiu.edu
[5]abdullahm@uthm.edu.my
[6]nurhanifah@uinsu.ac.id
*Corresponding author

Received: 15/1/2025 Revised: 1/4/2025 Accepted: 13/4/2025 Published: 30/4/2025

ABSTRACT

This paper investigates the effect of gender and task complexity on software development performance, addressing a critical gap in empirical research and predictive modelling that warrants further investigation. Understanding how these factors interact is crucial for optimising task allocation and enhancing team efficiency in software engineering environments. To investigate this, data were collected from two universities involving 180 software development students with an equal representation of male and female participants. Participants were required to complete programming tasks at three levels of complexity—easy, medium, and hard—to assess their performance under varying cognitive loads. The results indicate a performance disparity based on task complexity, where male software developers consistently outperformed their female counterparts as the complexity of the tasks

increased. Further analysis using predictive modelling techniques, which are logistic regression and decision tree models, revealed that logistic regression consistently achieved higher accuracy and F1 and AUC-ROC scores, demonstrating its superior predictive capability in modelling performance variations. Despite these findings, further research is necessary to explore additional interventions to help optimise task allocation based on individual strengths and expertise. Additionally, future studies should continue to challenge stereotypes and explore innovative strategies for promoting diversity and equality in software development.

Keywords: Gender, software team, task complexity, team performance.

INTRODUCTION

In software engineering (SE), task performance is a key measure of individual and team productivity. Software development tasks vary widely in complexity and require technical expertise and strong cognitive and collaborative skills. As teams become increasingly diverse, it is essential to understand how different factors, such as gender, affect performance under varying tasks. Research has shown that men often exhibit systematic, analytical thinking, while women tend to use holistic, intuitive approaches (Philbin et al., 1995; Ceci et al., 2023). However, the impact of these cognitive differences and diversity on software development performance remains unclear complexities (Rodríguez-Pérez et al., 2021). Recent studies have begun exploring these dynamics in modern software teams. For instance, Saeter et al. (2024) found that gender diversity can moderate the relationship between task complexity and team performance in agile environments, while Sunder et al. (2024) provided empirical evidence that task complexity plays a critical role in determining the performance of gender-diverse software teams.

One of the key issues this study addresses is the lack of attention given to gender in research on personality and performance within software development. This gap demands a more inclusive and thoughtful approach—one that recognises the importance of gender diversity in how tasks are assigned and how performance is assessed. Overlooking how task complexity interacts with gender may unintentionally increase stress and hinder performance among developers. By understanding performance patterns linked to gender and task complexity, project managers can make more informed decisions in assigning tasks, leading to healthier work environments and improved team productivity.

This study investigates whether gender significantly affects the performance of software developers when task complexity is taken into account. Thus, grounded in the gender similarities hypothesis—which posits that men and women are more alike than different in cognitive abilities (Hyde, 2016)—the null hypothesis (H_0) asserts that gender does not significantly influence performance when task complexity is considered. Furthermore, this study explores the use of predictive modelling to identify suitable models for assessing software team performance based on gender and task complexity. The research aims to deepen our understanding of performance dynamics in diverse software engineering teams by addressing this gap.

RELATED WORKS

Research on the relationships between gender, task complexity, and performance in software development is complex and reveals diverse perspectives. Gender-related cognitive and behavioural differences, task complexity levels, and contextual factors such as collaboration settings and

organisational culture contribute to the performance dynamics in software development. This review critically examines existing literature on gender differences in cognitive styles, the influence of task complexity on performance, collaborative versus individual settings, and empirical studies on software developer performance. The review also highlights gaps and areas for further exploration, ultimately laying the foundation for the hypothesis that gender may not significantly impact performance under task complexity conditions.

Gender and Cognitive Styles

Cognitive styles refer to consistent ways of processing information, problem-solving, and decision-making. Traditional views have suggested that men tend to exhibit systematic, analytical thinking, whereas women often demonstrate holistic, intuitive approaches (Halpern, 2012). For example, men are typically associated with logical reasoning and precision, making them better suited to tasks requiring systematic methodologies, while women's intuitive problem-solving is argued to be advantageous in tasks demanding empathy and creativity. However, these generalisations are increasingly being challenged. Hyde's (2005) gender similarities hypothesis argues that men and women are more similar than different in cognitive abilities, with any variations primarily shaped by task-specific demands rather than inherent gender traits. In software development, tasks like debugging or algorithmic problem-solving often reveal minimal gender differences when factors such as experience and education are accounted for. It demonstrates that cognitive styles are shaped by situational and contextual factors rather than fixed gender-based distinctions.

Recent empirical studies have shown that individual differences such as expertise, motivation, and domain familiarity are more significant predictors of performance than gender (Suárez & Vizcaino, 2023; Lin & Wong, 2024). For example, Woo and Kim (2022) showed that no significant gender differences were found in debugging performance for moderately and highly complex scenarios when skill and prior experience were accounted for. In the same way, Sun et al. (2024) argued that there are small differences in programming tasks, but those differences are mostly resolved due to the extensive practice and training that people do. So, these results refute the assumption that gender is the main factor influencing task performance and instead focus on individual differences and the context.

Empirical research examining gender differences in software developer performance under varying task complexity conditions is limited but growing. Woo and Kim (2022) found no significant gender-based differences in programming tasks, emphasising the role of skill and experience over gender. Similarly, Sun et al. (2024) meta-analysis concluded that while programming tasks occasionally exhibit minor gender variations, these differences are negligible when accounting for training and educational background. Moreover, research by Schauer et al. (2025) highlighted the influence of task allocation practices on performance outcomes. Women were often assigned documentation or less complex tasks within software teams, which skewed performance evaluations and reinforced gender stereotypes. It demonstrates the importance of equitable task distribution to ensure accurate performance assessments.

The relationship between gender, task complexity, and performance in software development is complex and varied. While traditional views emphasise gender-based cognitive differences, contemporary research highlights these variations' situational and task-dependent nature, thus needing to be explored specifically in task complexity.

Task Complexity and Performance

Task complexity is a pivotal factor influencing performance, encompassing aspects such as problem difficulty, required skills, and cognitive load. High-complexity tasks in software development—such as designing complex systems or debugging large-scale applications—demand advanced problem-solving abilities and effective cognitive load management. In this technological world, technological advancement makes many tasks easy to perform, but at the same time, some other tasks create more complications than ever before (Rescher, 2019). There are two sides to a coin. Technological developments fix existing issues and create other problems for themselves (Liu & Li, 2012). Many studies have considered a negative relationship between task complexity and performance (Bravo et al., 2015), one of the most crucial performance factors (Liu & Li, 2012). It is the core issue in operating activities that employees face (Braarud & Kirwan, 2010). Wood (1986) believed that the complexity of tasks affects human performance and actions by putting burdens on the performers of tasks.

Jacko and Ward (1996) and Liu and Li (2012) claimed that the task's difficulty impacted the participant's mental activity and its effect on their performance. Similarly, research conducted by Timmermans (1993), Bonner (1994), and Chung and Monroe (2001) all concluded that high task complexity impacts decisions and decreases the judgment quality. Another thing Hou et al. (2019) mentioned is that task complexity affects the quality of task completion. It is significantly affected that when the complexity of the task is higher, the error rate in the task is higher while completing the task. Complex tasks allow subjects to process more information signals, perform more separate actions, and tackle more complex interaction requirements. Higher complexity is expected to increase the time required to perform a task (Topi et al., 2005). The result of Topi et al. (2005) was that increased complexity had a significant negative impact on all aspects of subjects' performance at each level of availability and independent of the strategy of speed/correctness. Wood (1986) claimed that in his study, a curvilinear system initially has a higher complexity and positively impacts performance. However, a higher level of complexity decreases the performance because the task requirements surpass the capacity of the task managers.

In the study of Saeedi (2020), he claimed that while performing any task, complexity and accuracy depend on the planning or given time. It shows that whenever the time is short, it increases the task's complexity and impacts the performance's accuracy. Remington gives some factors contributing to increasing the project complexity under the following headings: goals, stakeholders, interfaces and interdependencies, technology, management process, work practices, and time (Remington et al., 2009). The study by Gong et al. (2025) highlights that language models perform effectively on simple code optimisation tasks but face challenges with more complex scenarios requiring deep contextual understanding. Meanwhile, Zhang et al. (2025) demonstrate that alignment training enhances the performance of code generation models, particularly in handling complex software engineering tasks. Table 1 depicts a summary of related works on task complexity and performance.

Table 1

Summary Related Works on Task Complexity and Performance

Studies	Findings	Identified Research Gaps
Zhang et al. (2025)	Alignment training improves model performance in complex software engineering tasks.	It does not explore how gender diversity may influence performance or the effectiveness of alignment in diverse teams.
Gong et al. (2025)	Language models perform well on simple code optimisation tasks but struggle with complex tasks requiring deep context.	Lacks consideration of gender diversity and does not integrate inclusive task allocation strategies in performance modelling.
Saeedi (2020)	Accuracy depends on task planning and available time; short time increases complexity.	It does not incorporate demographic or role-based variables.
Hou et al. (2019)	Higher task complexity leads to increased error rates in task completion.	It does not consider adaptive strategies or team diversity.
Bravo et al. (2015)	Negative relationship between task complexity and performance.	Insufficient analysis in diverse team environments (e.g., gender roles).
Liu and Li (2012)	Technological developments often create new problems; task complexity negatively impacts performance.	Need for contextual understanding of complexity in modern software teams.
Remington et al. (2009)	Project complexity is influenced by goals, stakeholders, tech, processes, and time.	Broad framework; lacks empirical focus on individual/team performance in the software context.

Based on Table 1, task complexity is a major factor influencing performance, particularly when time and planning are constrained. Higher complexity has been linked to increased error rates and cognitive overload, especially in dynamic environments. However, across these studies, a common gap remains: limited integration of diversity factors—such as gender and task complexity—and a lack of predictive modelling that can guide task allocation in modern, diverse software teams. It highlights the need for updated, inclusive models that better reflect current development practices.

Predictive Modeling Techniques

This study employs predictive modelling techniques to examine the relationship between gender, task complexity, and software team performance. Logistic regression and decision tree techniques were selected due to the categorical nature of the data and their effectiveness in classification tasks. These methods allow for identifying patterns and generating interpretable models that support data-driven decision-making in software project performance and management.

Logistic Regression

The regression technique is appropriate for research purposes when researchers are planning to establish a relationship between variables to predict the results of variables. This technique can be applied efficiently and effectively as long as the researcher provides several data collection assumptions, such as multi-collinearity, data forming, linearity, data size, and data type. According to Dinesh and Kalyanasundaram (2023), logistic regression was the most effective machine learning method, surpassing Support Vector Machine (SVM), K-Nearest Neighbourhood (KNN), decision trees, and random forest in detecting breast cancer using the Wisconsin dataset. The study aimed to determine the accuracy of different techniques in diagnosing diseases, especially in identifying benign and malignant cells. These results indicate that, at least in the context of breast cancer detection, a predictive logistic regression model is efficacious and, thus, alongside other researchable aspects in cancer, may be an asset in medical prognostic research.

This technique is also utilised in the field of SE, bringing attention to the importance of the discipline in predicting and analysing data within the domain. Many researchers in SE adopted this technique as it is highly effective in dealing with multiple mixed data types predictor variables, yielding dependable binary outcomes (Feng et al., 2024; Ceran et al., 2023; Ibraigheeth & Fadzli, 2020). Moreover, this technique is proven to be the most effective algorithm in SE (Arya et al., 2019; Shen et al., 2019). The discussion from the previous research in which the researchers claimed that the logistic regression technique has been widely used, it would be fair and safe to assume that this technique is an effective and useful method for predicting binary results based on multivariate predictor variables.

Decision Tree

The decision tree is one of the most common classification methods widely used to construct a predictive model generated by data (Duran et al., 2023; Sarker et al., 2020). It has many benefits, such as ease of interpretation, the ability to manage multi-dimensional characteristics, speed and simplicity of design processing, acceptable accuracy of prediction, and the ability to produce human-comprehensible rules (Wu et al., 2016). Trees are portrayed as simple and easy phenomena under the umbrella of decision tree methodology, in which trees are represented in graphic forms that can be read and easily explained. Furthermore, the results of the analysis of the machine learning classification technique suggest that this technique may be superior in terms of output rendering when dealing with either categorical or discrete variables compared to neural networks, K-nearest neighbourhood (KNN), Naïve Bayes, and support vector machine (SVM) (Abuhaija et al., 2023; Kotsiantis, 2007).

To deal with categorical data, the researchers depend on the decision tree, which is profitable in examining mixed data in general and categorical data evaluation in particular. Nonetheless, this technique is often considered before being used because of its weakness in producing complex tree structures that cannot be easily incorporated into the data. Logistic regression is mainly used and is suitable for small sizes. Also, categorical data can help the researcher generate IF-THEN rules that can be easily understood. The main advantage of statistical logistic regression is that it is efficient when managing multiple mixed data type's predictors that produce accurate binary results. Thus, this study applied logistic regression and decision tree models, given the categorical nature of the data, and compared their performance using accuracy, F1-score, and Are Under Curve-Receiver Operating Characteristic (AUC-ROC) values.

METHODOLOGY

This study employed a quasi-experimental design to investigate the impact of gender and task complexity on software development performance. The methodology of the study is described by the following experimental design.

Experimental Design

A 2×2 factorial design was employed, with gender (male, female) as one factor and task complexity (easy, medium, hard) as the other. Participants were randomly assigned to work on individual software tasks that simulated real-world debugging and programming challenges. Task complexity was determined by lecturers using Bloom's Taxonomy and categorised as easy, medium, or hard based on the cognitive demands following the classification suggested by (Husić et al., 2019). The experiment was performed in three different stages. In the first stage, participants completed pre-questionnaires and signed consent forms before the sessions. The second stage included performing tasks and activities of the experiment in controlled conditions. In the last stage, participants completed post-questionnaires, after which the lecturer graded their performance on the tasks to assign performance marks. Each stage was planned to balance consistency and reliability across data collection and assessment. Every round of the experiment had different tasks, but the complexity level remained constant to minimise error or bias in the results.

The rationale for offering different programming tasks in two rounds of the experiment stemmed from a dual focus on internal validity and protecting against potential reactivity or testing effects. Considering the background of participants as university students with some programming knowledge, the possibility of behaviour changes due to task familiarity posed a significant risk. Maintaining the same task in both rounds would mean participants likely alter their approaches, compromising internal validity. The experiment was done in six rounds, each with the same set of tasks complexity, one for each participant in every round.

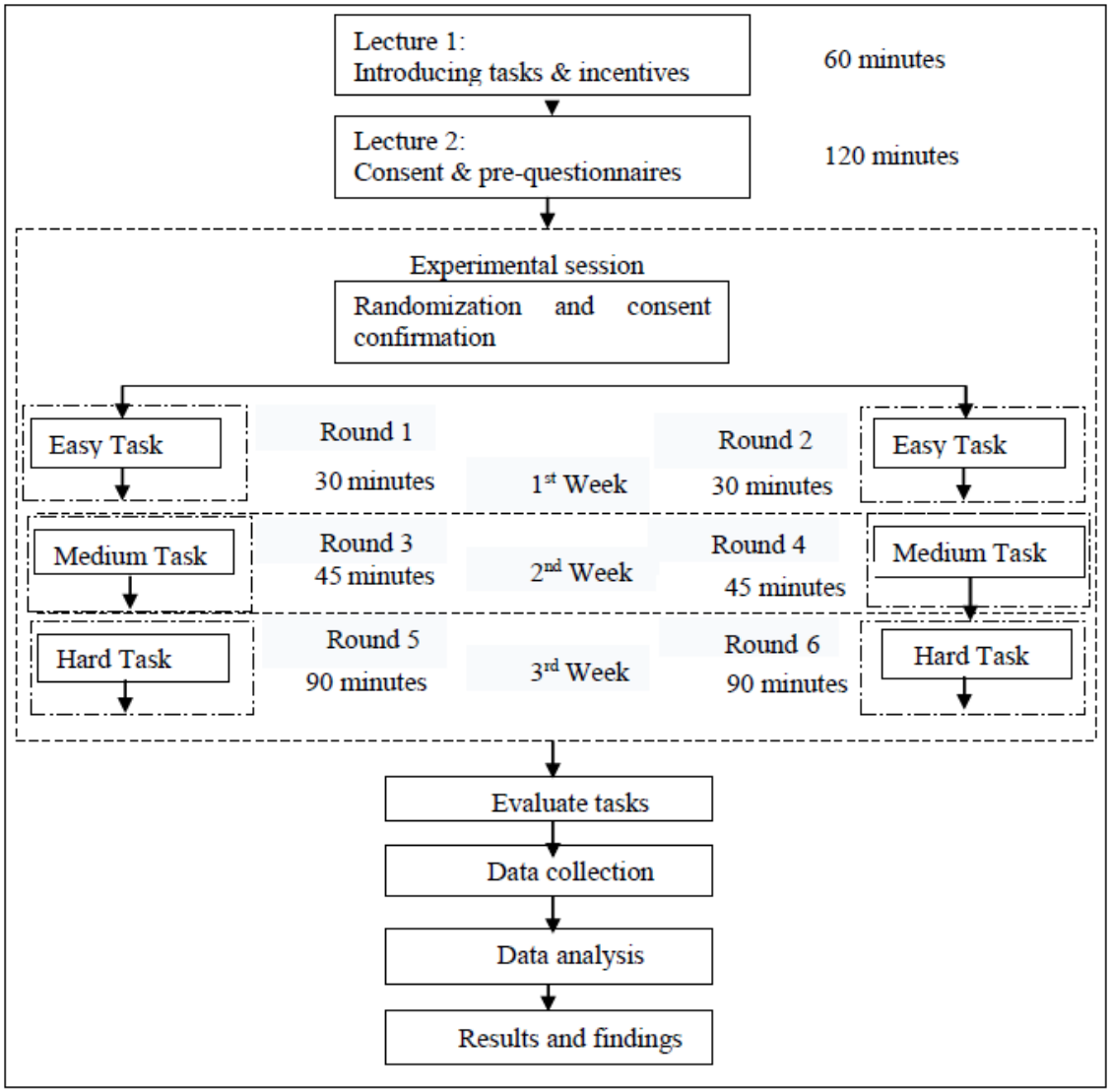
In explaining the protocols and requirements of the experiment, the first lecture was for one hour and was specially selected to ensure the participants understood their roles in the study. A meeting was conducted with the students in their usual class setting to discuss the experimental protocols. To encourage participants to take part in the experiments, an incentive was provided based on previous studies (Xue et al., 2021; Lo et al., 2020). According to their experiment performance, this incentive required teachers to add 10% to the student's final result. By doing so, the participants were expected to be motivated to take the study seriously and perform well (Ali & Anwar, 2021).To ensure that the experiment runs smoothly, the first lecture was sequenced and organised to present all of the experiment's terms and conditions to the participants so that they could easily meet the requirements. During the next hours of class, participants filled out all the forms and pre-questionnaires required before the experiment. However, it was important to verify the completion of these requirements before assigning participants to specific experiments. This approach aimed to facilitate a smooth and organised process, enabling participants to fully understand the experiment's expectations and complete the necessary documentation before proceeding further.

The experiments were divided into two level rounds during the first week of the experimental session. The experiment was designed to evaluate participants' performance across tasks of varying complexity and was conducted over two rounds for each complexity level. Two rounds per level were intended to produce average performance scores and ensure fair, consistent results. In the first week, participants

completed two rounds of tasks categorised as easy. Each task was estimated to take approximately 30 minutes, and participants were given ample time to complete them without constraints. The second task, conducted the following day, featured a similar level of complexity to maintain consistency and enable reliable comparison between rounds. In the second week, participants engaged in two rounds involving tasks of medium complexity, each estimated to require around 45 minutes. These sessions aimed to examine changes in performance and approach as task demands increased. Finally, in the third week, participants completed two rounds of high-complexity tasks, with an estimated duration of 90 minutes each. These tasks were designed to assess how participants coped with more cognitively demanding activities. During the experiment, all participants worked individually, and task conditions were kept consistent to ensure the reliability of observations related to task complexity. Figure 1 presents a flow chart illustrating the experimental design and how the experiments were conducted.

Figure 1

Flow Chart of Research Experimental Design



Participants and Sampling

The study employed purposive sampling, aligning with the research objective of evaluating software engineering team performance. Participants were recruited from two universities, specifically from programming classes, to ensure a balanced representation of male and female students with foundational software development skills. Demographic data (e.g., gender, years of programming experience, educational background) were collected using a pre-experiment questionnaire. Although using university students might initially appear to threaten external validity, previous research in software engineering has shown that student samples yield similar performance outcomes to professional developers (Feitelson, 2015; Falessi et al., 2017). Careful participant selection and experimental design ensured that the results were reliable and valid.

Task Procedures

Each participant completed a series of tasks under both low and high-complexity conditions. The complexity levels were defined as follows:

- **Easy:** Tasks requiring basic comprehension and application of simple programming concepts.
- **Medium:** Tasks involving moderate analysis and problem-solving skills.
- **Hard:** Tasks demanding synthesis and evaluation, requiring advanced debugging and algorithmic problem-solving skills.

Performance was measured using objective metrics evaluated by lectures based on code efficiency and correctness. The performance outcome was dichotomised into "effective" versus "ineffective" based on a task quality grade. The performance scores above 80 were categorised as effective class, while those below 80 were classified as ineffective. The threshold of 80% was established in alignment with agreements reached with the subject lecturer and grading scales. For instance, scores ranging from 80 to 84.9% corresponded to a grade of "A-" (equivalent to a 3.75 GPA), while scores of 85 and above were classified as grade "A" (equivalent to a 4.00 GPA). After task completion, a post-task survey was administered to capture participants' subjective perceptions of task difficulty and overall experience.

Data Collection and Analysis

Data was collected through pre- and post-questionnaires administered to the selected participants. The data collection process began with participants completing pre-questionnaires, consent forms, and bio-data forms. Throughout this process, the researchers took great care to maintain the confidentiality of the participants and their responses and assured them that their responses would only be used for research purposes and would not be shared with anyone outside of the research. Quantitative data collection from task performance based on task complexity and gender were analysed using binary logistic regression. The regression model estimated the probability of achieving high performance as a function of gender and task complexity while controlling for potential confounding factors such as prior programming experience.

Threats to Validity

To mitigate potential threats to validity, it is essential to carefully design and conduct experiments that minimise these concerns. According to Wohlin et al. (2012), there are four key types of validity threats in software engineering experiments.

Internal Validity

In this study, the entire experimental procedure was clearly explained to all participants, and everyone received identical instructions. No participants were divided into different groups, thereby eliminating risks of compensatory rivalry or participant disengagement. To prevent task familiarity from affecting results, distinct tasks were used in each round, with the only commonality being the level of task complexity. Participants were randomly assigned, and binary logistic regression was applied to control for individual differences, further enhancing internal validity.

Construct Validity

The tasks were systematically designed and categorised into easy, medium, and hard levels based on Bloom's Taxonomy to ensure construct validity. The complexity levels were validated by subject matter experts (university lecturers), and performance was evaluated using standardised university grading rubrics. Although the controlled nature of these tasks may not fully replicate real-world software development scenarios, prior research has shown that such experimental designs can provide valuable insights (Kohl & Prikladnicki, 2024; Schauer et al., 2025).

Conclusion Validity

Conclusion validity relates to the accuracy of inferences drawn from the data. This study was conducted systematically, and the tools and techniques used for measurement were selected based on a comprehensive literature review. Statistical assumptions were tested before hypothesis validation to ensure that the conclusions were scientifically sound. Furthermore, the experiments were conducted in a controlled computer lab environment, with continuous invigilation to minimise risks of plagiarism or external interference, thereby improving the reliability of the results.

External Validity

While the study involved participants from two university-level programming classes—potentially limiting generalizability—extensive research in software engineering education suggests that students can exhibit behaviour and performance patterns similar to professional developers in experimental contexts (Feitelson, 2015; Falessi et al., 2017). It supports the broader relevance of the findings beyond the academic setting.

Ethical Considerations

All participants were informed about the study's purpose and procedures, and their informed consent was obtained before participation. Confidentiality and anonymity were maintained throughout the data collection and analysis process.

RESULTS

This study primarily examines the participants' performance, which is the dependent variable. This performance has been categorised into two classes: effective and ineffective performance. The performance scores above 80 were categorised as effective class, while those below 80 were classified as ineffective class, as described in the methodology section. A total of 180 data sets were successfully

collected. The following sections analyse key factors influencing performance under different task complexities. Table 2 presents the classification of participants' performance across three task complexity levels: easy, medium, and hard.

Table 2

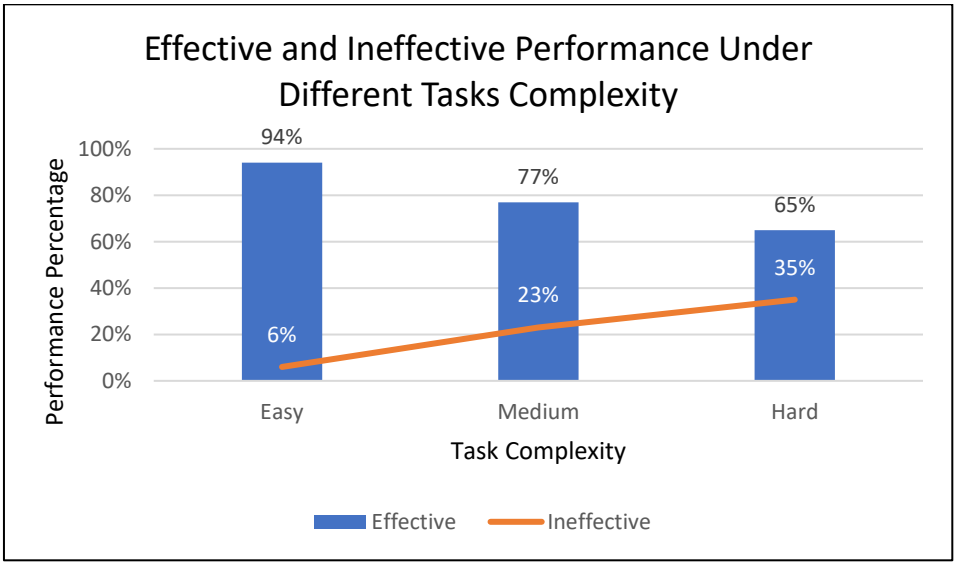
Total Number of Effective and Ineffective Performances

Dataset	Performance (n=180)					
	Effective			Ineffective		
A & B	Easy	Medium	Hard	Easy	Medium	Hard
	169	139	116	11	41	64

Table 2 presents Dataset A and Dataset B, which consist of data collected from two universities. The table categorizes participants' outcomes across three levels of task complexity—easy, medium, and hard—based on their performance in distinct task rounds. This table provides an overview of the number of participants who performed at each complexity level and the classification of their performance as effective or ineffective based on the empirical results from the dataset during their final task evaluation. Furthermore, the dataset results are graphically represented in Figure 2, highlighting an interesting pattern in the data.

Figure 2

Effective and Ineffective Performance



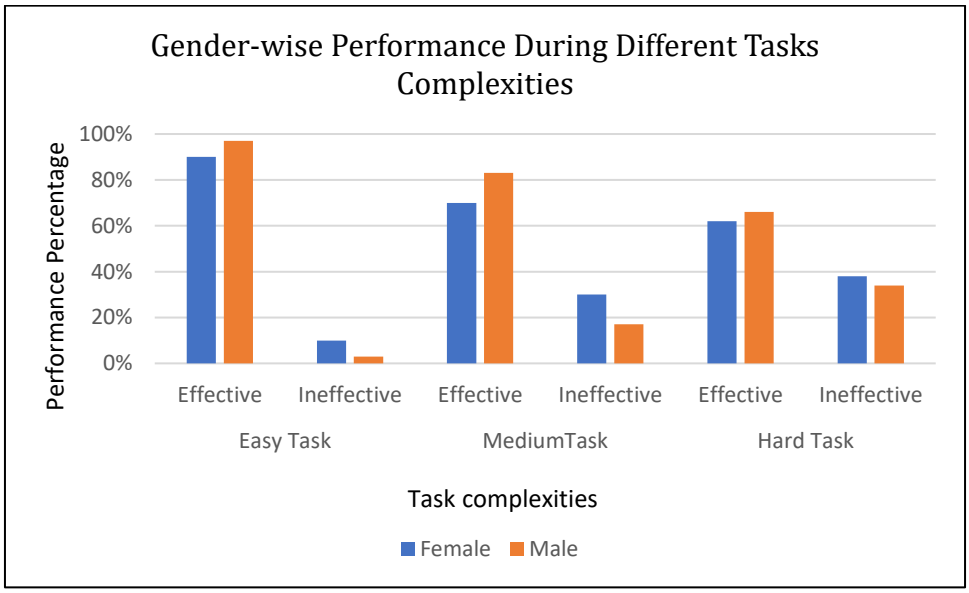
Both Table 2 and Figure 2 reveal an interesting pattern: as task complexity increases, performance declines. Specifically, the number of participants achieving effective performance diminishes as they progress from easy to medium and then hard tasks. This pattern suggests that increased complexity imposes greater cognitive demands on participants, affecting their ability to perform effectively. Task complexity influences performance because higher complexity requires more cognitive resources, such as problem-solving skills, memory retention, and attention span. As complexity increases, participants may struggle to maintain focus, leading to a higher likelihood of ineffective performance. This observation underscores the importance of considering task complexity alongside other influencing factors, such as gender, when evaluating performance.

Influence of Gender on Task Performance

The participants' demographic profiles were carried out from the SE participants. The participants in this study were all between the ages of 20 and 25. It is because all participants were from the SE department, and their ages were nearly similar. There were 103 males and 77 females among the 180 participants, for a ratio of 57% and 43% percent, respectively. The analysis reveals significant patterns in the relationship between gender, task complexity, and performance effectiveness. Figure 3 shows males' and females' effective and ineffective performances under different task complexities.

Figure 3

Gender-wise Performance During Different Tasks Complexities



According to the results, both males and females demonstrated effective performance, with males generally outperforming females across all task complexities. Males achieved 97% effectiveness for easy tasks, while females achieved 90%. Males maintained 83% effectiveness in medium tasks compared to 70% for females. In hard tasks, male performance dropped to 66%, whereas females achieved 62%. As task complexity increased, both genders experienced a decline in performance, but the impact was more pronounced among female participants. These findings suggest that task complexity significantly influences performance, with a noticeable gender-based variation in effectiveness.

It is evident that both genders perform worse on tasks with increasing complexity, but the effect seems to be more noticeable for female participants. The performance gap between males and females increases on medium-complexity tasks, indicating that females have a harder time adjusting to increasing complexity than males. This discrepancy may be due to differences in cognitive processing strategies, approaches to problem-solving, or confidence levels when taking on more difficult tasks. Moreover, the decreasing performance pattern for both genders implies that increased task complexity places large cognitive demands on participants. These demands probably call for improved memory recall, decision-making, and problem-solving skills, which might affect overall efficacy. The fact that both males and females showed a consistent decline in performance reinforces the notion that task complexity is crucial in determining success rates, regardless of gender.

These results imply that task complexity substantially impacts performance outcomes, with a discernible difference in efficacy by gender. Understanding these differences can provide valuable insights for designing task structures, training programs, and performance evaluation criteria in professional and educational settings. Future studies could examine the cognitive processes that underlie these gender-based differences and methods to close performance gaps and improve task flexibility for both men and women.

Hypothesis Testing: Moderation of Gender

The study's hypothesis aims to demonstrate how gender influences how software developers handle tasks of varying complexity. The following hypothesis was tested:

$H1_0$: There is no significant moderation by gender (male and female) on the effect of task complexity on software developers' performance.
$H1_A$: There is a significant moderation by gender (male and female) on the effect of task complexity on software developers' performance.

A binary logistic regression analysis examined the relationship between gender, task complexity, and software developers' performance. The omnibus test of model coefficients indicated a significant overall association between the variables, $\chi^2(3) = 36.278$, $p < .001$. The logistic regression model, which included an interaction term representing the combined effect of task complexity and gender, demonstrated a good fit to the data, $\chi^2(3) = 22.765$, $p < .001$. The model explained a notable proportion of variance in performance, as reflected by Cox & Snell $R^2 = 0.065$ and Nagelkerke $R^2 = 0.092$.

The main effect of gender was statistically significant ($B = 0.621$, $p = .036$), suggesting that gender alone had a meaningful impact on software developers' performance. The odds ratio for gender was 1.855, indicating that male developers had higher odds of effective performance than female developers across tasks. Additionally, the interaction term for task complexity and gender was a significant predictor, $\chi^2(2) = 27.489$, $p < .001$, reinforcing the idea that the effect of task complexity on performance varied by gender.

These findings suggest that gender moderates the relationship between task complexity and software developers' performance, leading to rejecting the null hypothesis ($H1_0$). The results indicate that male and female developers respond differently to increasing task complexity, with male developers generally maintaining higher performance levels across tasks. In conclusion, these results highlight the importance of considering gender-specific differences in performance when assigning software development tasks of varying complexity.

Modeling for Performance Prediction

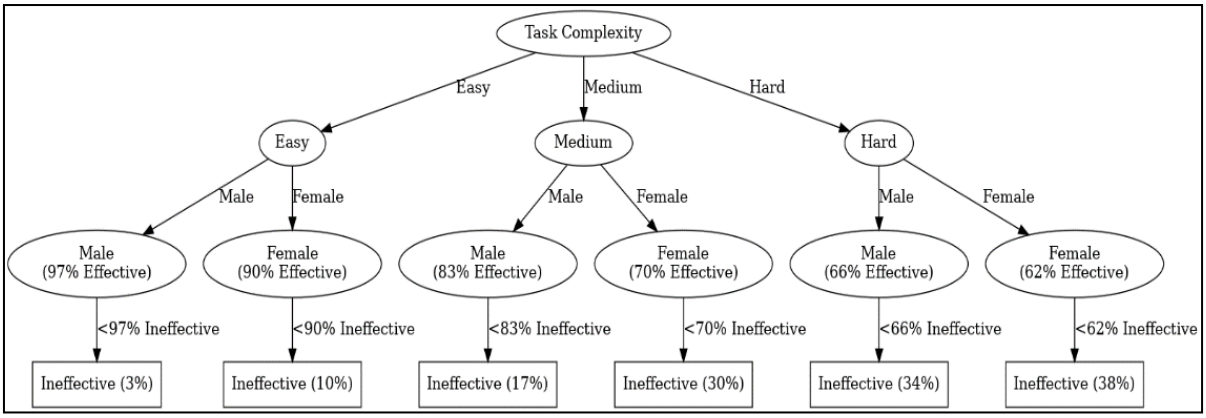
The analysis was extended to compare the accuracy levels of different models. Two predictive modelling techniques were applied to assess the impact of gender and task complexity on software team performance: Binary Logistic Regression and Decision Tree. These techniques were chosen to assess predictive accuracy, precision, recall, F1-score, and AUC-ROC across three task complexity levels (Easy, Medium, and Hard). These methods are widely used in predictive modelling and data mining for classification problems. Logistic regression, as a statistical modelling approach, helps understand the relationship between gender, task complexity, and performance by estimating probabilities. In contrast, Decision Trees classify data by learning hierarchical rules, visually representing decision-making patterns. Comparing these techniques allows us to determine which better predicts software developers' performance under different levels of task complexity.

Decision Tree Analysis

Figure 4 illustrates the decision tree representing performance classification based on task complexity and gender. The tree follows a hierarchical structure, where task complexity serves as the root node, branching into gender categories and further classifying individuals as effective or ineffective based on their performance.

Figure 4

Decision Tree of Performance Based on Task Complexity and Gender



The decision tree illustrates the relationship between task complexity, gender, and performance effectiveness in software engineering teams. At the root level, task complexity is divided into easy, medium, and hard tasks, which reflects the growing cognitive and problem-solving demands developers face. From this starting point, the tree branches into gender-based divisions, which separate male and female participants to evaluate how performance outcomes vary across different difficulty levels. Each gender-based category is further divided into performance classifications, which determine whether developers are effective or ineffective based on their capacity to complete tasks successfully.

The decision tree's hierarchical structure shows a recurring trend: the percentage of ineffective performance rises as task complexity rises. This pattern draws attention to the increasing cognitive load and difficulties in solving problems with increasingly challenging tasks. Despite performance drops for both male and female developers, gender-based disparities are still noticeable at all levels of complexity. While female developers see a more noticeable decline in effectiveness, especially in medium and difficult assignments, male developers typically exhibit greater adaptability in complicated settings. These differences imply that experience levels and cognitive workload management may affect performance as task difficulty increases.

Software teams and project managers can optimise task allocation strategies by mapping these insights into a structured decision tree. By understanding the relationship between gender and task complexity, specific training programs, skill enhancement initiatives, and support mechanisms can be developed to ensure that all developers can perform efficiently, regardless of the task's complexity.

Logistic Regression Model for Performance Classification

Logistic regression is a statistical model used for binary classification, meaning it predicts the probability of an outcome belonging to one of two categories. In this study, it classifies performance as either Effective (1) or Ineffective (0) based on two independent variables:

- Task Complexity (*Easy, Medium, Hard*)
- Gender (*Male, Female*)

The logistic regression equation is expressed as in Equation 1:

$$\text{Log}\left(\frac{P(effective)}{1-P(effective)}\right) = \beta 0+\beta 1(\text{Gender})+\beta 2(\text{Task Complexity}) \quad (1)$$

Where:

- P(Effective) is the probability of effective performance.
- β0 is the intercept.
- β1, β2 are the coefficients that determine how much gender and task complexity impact performance.

Logistic Regression Results

Based on the study, the estimated logistic regression is:

$$\text{Log}\left(\frac{P(effective)}{1-P(effective)}\right) = -1.1801+0.4714\cdot\text{Gender}-0.4599\cdot\text{Task Complexity}$$

To convert the log odds into an actual probability, we use the logistic function as in Equation 2:

$$P=\frac{e^{(-1.1801+0.4714\cdot\text{Gender}-0.4599\cdot\text{Task Complexity})}}{1+e^{(-1.1801+0.4714\cdot\text{Gender}-0.4599\cdot\text{Task Complexity})}} \quad (2)$$

Equation 2 predicts the likelihood of a participant being effective based on gender and task complexity level. The intercept (-1.1801) represents the baseline log odds of being effective when the female participant faces easy tasks. The gender coefficient (0.4714) suggests that being male increases the log odds of effective performance. However, this effect is not statistically significant, implying that gender alone does not strongly predict performance. The task complexity coefficient (-0.4599) indicates that as task complexity increases, the probability of effective performance decreases, with harder tasks leading to a higher likelihood of ineffective performance.

By plugging in values for gender and task complexity, this model can estimate the probability of a participant being effective in a given scenario. For example, a male participant facing a medium task will have a different probability of effectiveness than a female facing a hard task, helping to quantify how gender and task difficulty interact in predicting performance outcomes.

Classification Results for Decision Tree and Logistic Regression

Table 3 depicts the classification results for easy, medium, and hard tasks using logistic regression and decision tree across all task complexities.

Table 3

Classification Results

Task Complexity	Logistic Regression Accuracy	Decision Tree Accuracy	Logistic Regression F1-score	Decision Tree F1-score	Logistic Regression AUC-ROC	Decision Tree AUC-ROC
Easy	86.9%	78.1%	86.8%	80.9%	91.1%	81.7%
Medium	93.4%	77.3%	90.4%	79.0%	95.6%	81.8%
Hard	88.3%	81.1%	85.1%	74.1%	96.2%	86.2%

The results indicate that logistic regression consistently outperforms decision trees in terms of accuracy, F1-score, and AUC-ROC across all task complexities. It suggests that logistic regression is a more effective model for predicting performance variations in software development tasks. The superior AUC-ROC scores highlight that Logistic Regression provides better discrimination between high and low-performance levels, making it a more robust approach for understanding how gender and task complexity influence performance. Furthermore, the reliability of logistic regression in predictive modelling for software teams may be attributed to the categorical nature of the dataset, as this method is well-suited for handling binary and ordinal variables. Hence, logistic regression is a more suitable classification approach for analysing performance variations in software development tasks.

DISCUSSIONS

The results of the classification show distinct differences in performance by gender and task complexity: both male and female software developers are highly effective in easy tasks, indicating that task complexity has little effect on performance at this stage. This suggests that both groups can easily perform easy tasks, probably because they are familiar with them and have lower cognitive demands. Though at different rates, both genders show a gradual decline in performance as task complexity increases, with male developers maintaining a more stable performance in medium-complexity tasks and female developers experiencing a more noticeable drop in effectiveness. This discrepancy suggests that increased task complexity presents additional cognitive challenges for female developers, possibly due to different problem-solving approaches, prior experience, or strategies for managing cognitive load.

For hard tasks, the performance gap between male and female developers narrows slightly but remains statistically significant, with male developers continuing to achieve higher accuracy. This trend suggests that while both genders face difficulties as complexity rises, male developers demonstrate greater adaptability in high-pressure, challenging scenarios. The ability to adjust to demanding tasks more effectively may stem from differences in training, experience levels, or confidence in problem-solving

strategies. Both male and female software developers demonstrate high performance in easy tasks, indicating that task difficulty does not significantly impact their effectiveness at this level. However, as task complexity increases, performance declines for both genders at different rates**.** Male developers maintain relatively stable performance in medium-complexity tasks, whereas female developers exhibit a more noticeable drop in effectiveness. It suggests that task complexity introduces additional challenges for female developers, potentially due to factors such as cognitive load or prior experience. The performance gap narrows slightly for the hard tasks but remains significant, with male developers still achieving better accuracy. This trend implies that while both genders struggle with increased complexity, male developers demonstrate greater adaptability under challenging scenarios.

These findings highlight the significance of taking gender differences into account in addition to task complexity when assessing software developer's performance. By identifying these patterns, software teams can optimise task assignments, offer targeted support mechanisms, and create strategies to improve overall productivity. By recognising how complexity affects different people, organisations can strive to create a more effective and inclusive work environment that helps developers realise their full potential.

CONCLUSION

The relationship between gender, task complexity, and performance in software development is complex and varied. While traditional views emphasise gender-based cognitive differences, contemporary research highlights these situational and task-dependent variations. Empirical evidence emphasises the role of individual attributes, such as expertise and motivation, over gender as predictors of performance. Furthermore, the influence of implicit biases and the importance of equitable task allocation cannot be overlooked. Organisations can create inclusive environments that foster equal opportunities by focusing on individual capabilities and addressing systemic biases. Future research should focus on supporting all developers, regardless of gender, in overcoming task complexity through inclusive learning environments and tailored interventions**.** Challenging stereotypes and promoting diversity can foster more innovative, balanced, and resilient software engineering teams**.**

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Abuhaija, B., Alloubani, A., Almatari, M., & Jaradat, G. M. (2023). A comprehensive study of machine learning for predicting cardiovascular disease using Weka and SPSS tools. *International Journal of Electrical and Computer Engineering*, *13*(2), 1891–1902. https://doi.org/10.11591/ijece.v13i2.pp1891-1902

Ali, B. J., & Anwar, G. (2021). An empirical study of employees' motivation and its influence job satisfaction. *International Journal of Engineering, Business and Management*, *5*(2), 21–30. https://doi.org/10.22161/ijebm.5.2.3

Arya, D., Wang, W., Guo, J. L. C., & Cheng, J. (2019). Analysis and detection of information types of open source software issue discussions. In *Proceedings of the International Conference on Software Engineering* (pp. 454–464). IEEE. https://doi.org/10.1109/ICSE.2019.00058

Braarud, P. Ø., & Kirwan, B. (2011). Task complexity: What challenges the crew and how do they cope? In A. B. Skjerve & A. Bye (Eds.), *Simulator-based human factors studies across 25 years: The history of the Halden Man-machine Laboratory* (pp. 233–251). Springer. https://doi.org/10.1007/978-0-85729-003-8_15

Bonner, S. E. (1994). A model of the effects of audit task complexity. *Accounting, Organizations and Society, 19*(3), 213–234. https://doi.org/10.1016/0361-3682(94)90033-7

Bravo, E. R., Santana, M., & Rodon, J. (2014). Information systems and performance: The role of technology, the task and the individual. *Behaviour & Information Technology, 34*(3), 247–260. https://doi.org/10.1080/0144929X.2014.934287

Ceci, S. J., Kahn, S., & Williams, W. M. (2023). Exploring gender bias in six key domains of academic science: An adversarial collaboration. *Psychological Science in the Public Interest, 24*(1), 15–73. https://doi.org/10.1177/15291006231163179

Ceran, A. A., Ar, Y., Tanrıöver, Ö. Ö., & Seyrek Ceran, S. (2023). Prediction of software quality with machine learning-based ensemble methods. *Materials Today: Proceedings, 81*, 18–25. https://doi.org/10.1016/j.matpr.2022.11.229

Chung, J., & Monroe, G. S. (2001). A research note on the effects of gender and task complexity on an audit judgment. *Behavioral Research in Accounting, 13*(1), 111–125. https://doi.org/10.2308/bria.2001.13.1.111

Dinesh, P., & Kalyanasundaram, P. (2023). Medical image prediction for the diagnosis of breast cancer and comparing machine learning algorithms: SVM, logistic regression, random forest and decision tree to measure accuracy of prediction. *AIP Conference Proceedings, 2821*(1), 050001. https://doi.org/10.1063/5.0158449

Duran, Z., Akargöl, İ., & Doğan, T. (2023). Data mining, Weka decision trees. *Orclever Proceedings of Research and Development, 3*(1), 401–416. https://doi.org/10.56038/oprd.v3i1.376

Falessi, D., Juzgado, N., Wohlin, C., Turhan, B., Münch, J., Jedlitschka, A., & Oivo, M. (2017). Empirical software engineering experts on the use of students and professionals in experiments. *Empirical Software Engineering, 23*, 452–489. https://doi.org/10.1007/s10664-017-9523-3

Feitelson, D. (2015). Using students as experimental subjects in software engineering research - A review and discussion of the evidence. *arXiv*, abs/1512.08409.

Feng, S., Keung, J., Xiao, Y., Zhang, P., Yu, X., & Cao, X. (2024). Improving the undersampling technique by optimising the termination condition for software defect prediction. *Expert Systems with Applications, 235*, 121084. https://doi.org/10.1016/j.eswa.2023.121084

Gong, J., Voskanyan, V., Brookes, P., Wu, F., Jie, W., Xu, J., Giavrimis, R., Basios, M., Kanthan, L., & Wang, Z. (2025). Language models for code optimisation: Survey, challenges and future directions. *arXiv preprint arXiv:2501.01277*. https://doi.org/10.48550/arXiv.2501.01277

Halpern, D. F. (2012). *Sex differences in cognitive abilities* (4th ed.). Psychology Press. https://doi.org/10.4324/9780203816530

Hou, W., Li, M., & Huang, J. (2019). Performance study of light assembly operations considering time pressure and task complexity. *DEStech Transactions on Computer Science and Engineering*, CSCME. https://doi.org/10.12783/dtcse/cscme2019/32563

Hu, L. (2024). Exploring gender differences in computational thinking among K-12 students: A meta-analysis investigating influential factors. *Journal of Educational Computing Research, 62*(5), 1358–1384. https://doi.org/10.1177/07356331241240670

Husić, J. B., Alagić, E., Baraković, S., & Mrkaja, M. (2019). The influence of task complexity and duration when testing QoE in WebRTC. In *Proceedings of the 18th International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1–6). IEEE. https://doi.org/10.1109/INFOTEH.2019.8717657

Hyde, J. S. (2005). The gender similarities hypothesis. *American Psychologist, 60*(6), 581–592. https://doi.org/10.1037/0003-066X.60.6.581

Hyde, J. (2016). Sex and cognition: Gender and cognitive functions. *Current Opinion in Neurobiology, 38*, 53–56. https://doi.org/10.1016/j.conb.2016.02.007

Ibraigheeth, M. A., & Fadzli, S. A. (2020). Software project failures prediction using logistic regression modeling. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICCIS49240.2020.9257648

Jacko, J. A., & Ward, K. G. (1996). Toward establishing a link between psychomotor task complexity and human information processing. *Computers & Industrial Engineering, 31*(1–2), 533–536. https://doi.org/10.1016/0360-8352(96)00192-1

Kohl, K., & Prikladnicki, R. (2024). Gender diversity on software development teams: A qualitative study. In D. Damian, K. Blincoe, D. Ford, A. Serebrenik, & Z. Masood (Eds.), *Equity, diversity, and inclusion in software engineering*. Apress. https://doi.org/10.1007/978-1-4842-9651-6_11

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica, 31*(3), 249–268. https://www.informatica.si/index.php/informatica/article/view/148

Lin, S., & Wong, G. K. W. (2024). Gender differences in computational thinking skills among primary and secondary school students: A systematic review. *Education Sciences, 14*(7), 790. https://doi.org/10.3390/educsci14070790

Liu, P., & Li, Z. (2012). Task complexity: A review and conceptualisation framework. *International Journal of Industrial Ergonomics, 42*(6), 553–568. https://doi.org/10.1016/j.ergon.2012.09.001

Lo, S. K., Lu, Q., Wang, C., Paik, H.-Y., & Zhu, L. (2021). A systematic literature review on federated machine learning: From a software engineering perspective. *ACM Computing Surveys, 54*(5), Article 95. https://doi.org/10.1145/3450288

Marchesi, M., & Gregory, P. (Eds.). (2024). *Agile processes in software engineering and extreme programming. XP 2024*. Lecture Notes in Business Information Processing, 512. Springer. https://doi.org/10.1007/978-3-031-61154-4

Philbin, M., Meier, E., Huffman, S., & Boverie, P. (1995). A survey of gender and learning styles. *Sex Roles, 32*, 485-494. https://doi.org/10.1007/BF01544184

Rodríguez-Pérez, G., Nadri, R., & Nagappan, M. (2021). Perceived diversity in software engineering: A systematic literature review. *Empirical Software Engineering, 26*, 102. https://doi.org/10.1007/s10664-021-09992-2

Remington, K., Zolin, R., & Turner, R. (2009). A model of project complexity: Distinguishing dimensions of complexity from severity. *Proceedings of the 9th International Research Network of Project Management Conference*, 11–13. https://eprints.qut.edu.au/29011/1/c29011.pdf

Rescher, N. (2019). *Complexity: A philosophical overview*. Routledge. https://doi.org/10.4324/9780429336591

Saeedi, M. (2020). Task condition and L2 oral performance: Investigating the combined effects of online planning and immediacy. *International Journal of Foreign Language Teaching & Research, 8*(32), 35–48. https://journals.iau.ir/article_674718_2ed8941322fb300c17788fc034f9f1cc.pdf

Saeter, G. E., Stray, V., Almås, S., & Lindsjørn, Y. (2024). The role of team composition in agile software development education: A gendered perspective. In D. Šmite, E. Guerra, X. Wang, M. Marchesi, & P. Gregory (Eds.), *Agile processes in software engineering and extreme programming* (Lecture Notes in Business Information Processing, Vol. 512, pp. 1–16). Springer. https://doi.org/10.1007/978-3-031-61154-4_12

Sarker, I. H., Colman, A., Han, J., Khan, A. I., Abushark, Y. B., & Salah, K. (2020). BehavDT: A behavioral decision tree learning to build user-centric context-aware predictive model. *Mobile Networks and Applications, 25*(3), 1151–1161. https://doi.org/10.1007/s11036-019-01443-z

Schauer, A. M., Liu, J., Saldaña, C., et al. (2025). Internal and external influences on role stereotype adherence and gender dynamics on engineering design teams. *International Journal of STEM Education, 12*, 3. https://doi.org/10.1186/s40594-025-00528-4

Shen, J., Baysal, O., & Shafiq, M. O. (2019). Evaluating the performance of machine learning sentiment analysis algorithms in software engineering. In *Proceedings of the IEEE 17th International Conference on Dependable, Autonomic and Secure Computing (DASC), IEEE 17th International Conference on Pervasive Intelligence and Computing (PiCom), IEEE 5th International Conference on Cloud and Big Data Computing (CBDCom), and 4th Cyber Science and Technology Congress (CyberSciTech)* (pp. 1023–1030). IEEE. https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00185

Suárez, J., & Vizcaíno, A. (2023). Stress, motivation, and performance in global software engineering. *Journal of Software: Evolution and Process, 35*(7), e2600. https://doi.org/10.1002/smr.2600

Sun, C., Yang, S., & Becker, B. (2024). Debugging in computational thinking: A meta-analysis on the effects of interventions on debugging skills. *Journal of Educational Computing Research, 62*(4), 1087–1121. https://doi.org/10.1177/07356331241227793

Sunder, M. V., Gangwar, M., & Modukuri, S. (2024). Do gender-diverse teams deliver better operational performance: An experimental study. *Production and Operations Management, 0*(0). https://doi.org/10.1177/10591478241260723

Timmermans, D. (1993). The impact of task complexity on information use in multi-attribute decision making. *Journal of Behavioral Decision Making, 6*(2), 95–111. https://doi.org/10.1002/bdm.3960060203

Topi, H., Valacich, J. S., & Hoffer, J. A. (2005). The effects of task complexity and time availability limitations on human performance in database query tasks. *International Journal of Human-Computer Studies, 62*(3), 349–379. https://doi.org/10.1016/j.ijhcs.2004.10.003

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer. https://doi.org/10.1007/978-3-642-29044-2

Woo, H., & Kim, J.-M. (2022). Impacts of learning orientation on the modeling of programming using feature selection and XGBOOST: A gender-focused analysis. *Applied Sciences, 12*(10), 4922. https://doi.org/10.3390/app12104922

Wood, R. E. (1986). Task complexity: Definition of the construct. *Organisational Behavior and Human Decision Processes, 37*(1), 60–82. https://doi.org/10.1016/0749-5978(86)90044-0

Wu, C.-C., Chen, Y.-L., Liu, Y.-H., & Yang, X.-Y. (2016). Decision tree induction with a constrained number of leaf nodes. *Applied Intelligence, 45*(3), 673–685. https://doi.org/10.1007/s10489-016-0785-z

Xue, S., Shi, X., Jiang, R., Feliciani, C., Liu, Y., Shiwakoti, N., & Li, D. (2021). Incentive-based experiments to characterise pedestrians' evacuation behaviors under limited visibility. *Safety Science, 133*, 105013. https://doi.org/10.1016/j.ssci.2020.105013

Yehuda, M., Manuel, A., & Imanuel, F. (2024). The effect of job pressure and task complexity on performance with resilience ability as moderator. *Primanomics: Jurnal Ekonomi & Bisnis, 22*(3), 37–51. https://doi.org/10.31253/pe.v22i3.2778

Zhang, K., Zhang, H., Li, G., You, J., Li, J., Zhao, Y., & Jin, Z. (2025). SEAlign: Alignment training for software engineering agent. *arXiv preprint*, arXiv:2503.18455. https://doi.org/10.48550/arXiv.2503.18455