Innovative Coding Teaching Methodologies: A Comprehensive Approach for Diverse Learners

[1]Pankaj Beldar*, [2]Rahul Rakhade, [3]Milind Bahiram, [4]Dr. Snehal Kadbhane
[1, 2, 3, 4] Department of Mechanical Engineering, K.K.Wagh Institute of Engineering Education and research, Savitribai Phule Pune University,
[1]prbeldar@kkwagh.edu.in ,[2]rdrakhade@kkwagh.edu.in, [3]mrbahiram@kkwagh.edu.in,
[4]svkadbhane@kkwagh.edu.in

Abstract—**This study investigates the pivotal role of coding skills in the education of mechanical engineering students, particularly in areas like data analytics, statistics, and numerical methods. It addresses challenges educators face, such as the abstract nature of programming and diverse student backgrounds, and explores innovative teaching methods such as pair programming and live coding sessions. By equipping students with coding proficiency, the study demonstrates how they can translate complex mathematical algorithms into practical simulations, crucial for analysing mechanical systems like computational fluid dynamics and finite element analysis. The research evaluates the effectiveness of these teaching strategies based on student feedback, emphasizing the need for personalized and innovative approaches to integrate coding effectively into the mechanical engineering curriculum. These insights aim to enhance coding education, preparing students to tackle complex engineering problems, make data-driven decisions, and thrive in today's technological landscape. The study contributes practical implications for educators seeking to bridge the gap between theoretical coding knowledge and its practical application in mechanical engineering, thereby preparing future engineers for the challenges of modern engineering practices.**

Keywords— **Flipped Classroom, Coding, Problem Solving, Numerical Methods and Statistics, Data Analytics.**
JEET Category—**Practice**

I. INTRODUCTION

Proficiency in coding holds immense significance for mechanical engineering students, especially concerning numerical methods, statistics, and data analytics.

In the realm of numerical methods, coding enables engineers to translate complex mathematical algorithms into practical simulations, facilitating the analysis of intricate mechanical systems and structures. Through coding, students can efficiently solve differential equations, model physical phenomena, and predict system behaviors, crucial for tasks like finite element analysis and computational fluid dynamics(Mudenagudi et al., 2019).

Moreover, in the domain of statistics, coding empowers mechanical engineers to process and analyze vast datasets, extracting meaningful insights that inform crucial engineering decisions. By applying statistical techniques through coding, students can assess the reliability of experimental results, optimize manufacturing processes, and enhance the overall quality of mechanical systems(Ghosh et al., 2020). Additionally, in the rapidly evolving field of data analytics, coding skills are indispensable. Mechanical engineers proficient in coding can harness the power of machine learning algorithms and statistical models to interpret complex data patterns, predict equipment failures, optimize energy consumption, and design innovative products. In essence, coding equips mechanical engineering students with the computational tools necessary to tackle intricate numerical problems, make data-driven decisions, and drive innovation within the industry, ensuring their readiness to meet the challenges of the modern engineering landscape(Bhoga et al., 2024). Teaching coding to mechanical engineering students presents challenges due to varying backgrounds, the abstract nature of programming, rapid technological changes, and the need for practical application. Integrating coding into mechanical projects, balancing theory with practice, and fostering problem-solving skills are ongoing hurdles. Educators must adapt their teaching methods to inspire students and bridge the gap between abstract coding concepts and real-world engineering applications. Teaching coding to mechanical engineering students poses several challenges. One major difficulty is the diverse backgrounds of students; some have prior programming experience, while others do not, leading to a disparity in skill levels within the same class. Additionally, the abstract nature of coding concepts can be daunting, especially for students more inclined towards hands-on engineering tasks. Keeping the curriculum engaging and relatable to mechanical engineering principles is a constant struggle(Yalagi et al., 2020). The fast-paced evolution of programming languages and tools demands constant updates to teaching materials, creating a challenge for educators to stay current. Moreover, integrating coding seamlessly into a mechanical engineering curriculum, ensuring its relevance and practicality, requires careful planning and coordination. Encouraging students to think algorithmically and fostering problem-solving skills in the context of coding add another layer of complexity(Arulkumar et al., 2022). Overcoming these challenges necessitates innovative teaching methods, personalized approaches, and a focus is on bridging the gap between theoretical aspects of coding and its application in the mechanical engineering field(Jamila, 2020). The code teaching methodologies discussed in this paper are Interactive Online Platforms, Live Coding Sessions, Pair Programming, Hackathons and Coding Competitions, Flipped Classroom Model, Interactive Coding Environments, Multimedia Learning, Real-World Projects, Peer-to-Peer Teaching, Storytelling and Scenarios, Incorporate IoT and Robotics, Online Coding Communities, Reflective Learning. Students in the third year of mechanical engineering taught numerical methods and statistics, while students in the final year of mechanical engineering were taught data analytics. To determine the outcomes of these techniques, honest feedback was obtained from the students. From the feedback analysis, the effectiveness of these techniques and student satisfaction was tested.

II. Literature Review

In the realm of education, the integration of extracurricular activities as a means to promote reflective learning has garnered attention (Díaz-Iso et al., 2019). Such activities serve as platforms for students to apply theoretical knowledge to practical scenarios, fostering deeper understanding and critical thinking skills. Reflective learning emerges as a crucial component in various educational contexts, including online courses (Weng et al., 2022), professional development for school leaders (Sharmin et al., 2019) and specialized fields like anaesthesiology (Goy et al., 2022). Reflective learning practices are not confined to traditional education settings but extend to diverse domains such as nursing (Naicker & Rensburg, 2018) and innovation communities (Dahlgaard & Dahlgaard-Park, 2006), These practices facilitate knowledge construction and transformative learning experiences, enabling individuals to adapt to evolving challenges effectively. Innovative methodologies, such as the community of inquiry framework and conversational functions for knowledge building communities, underscore the importance of critical thinking and collaborative learning environments (Kaczkó & Ostendorf, 2023); (Cacciamani et al., 2018)). These frameworks provide structured approaches to enhance cognitive presence and facilitate meaningful interactions among learners. Technological advancements have further influenced learning paradigms, with studies exploring the motivation of users in digital learning environments (Huang et al., 2012), Additionally, online training platforms have emerged to support skill development, as evident in the case of community therapists (Afzal et al., 2021). Beyond traditional educational spheres, interdisciplinary research sheds light on novel applications of technology in diverse sectors. For instance, IoT technology is leveraged in fields like agriculture (Micle et al., 2021), healthcare (Rahman et al., 2022) and urban planning (Hew & Lo, 2018), These studies highlight the multifaceted roles of technology in addressing real-world challenges and enhancing quality of life. Storytelling emerges as a compelling tool for scenario building and speculative thinking (Dowsett et al., 2022) By weaving narratives, individuals can explore diverse perspectives and anticipate future possibilities (Carbonell et al., 2017).This narrative approach transcends disciplinary boundaries, offering a holistic framework for envisioning and shaping the future. Peer-to-peer teaching has emerged as a prominent educational strategy across various disciplines, offering benefits such as enhanced learning outcomes and reduced costs. In healthcare settings, nurse-led initiatives employing peer-to-peer teaching have demonstrated effectiveness in reducing rates of catheter-associated urinary tract infections (CAUTIs) and related costs (Pashnik et al., 2017). Similarly, research in educational contexts suggests that peer-to-peer teaching influences learning outcomes and ability tracking (Kimbrough et al., 2022) with studies emphasizing its critical role in higher education (Stigmar, 2016). In healthcare, peer-to-peer teaching has been instrumental in facilitating interventions such as antibiotic stewardship, highlighting its efficacy in promoting knowledge transfer and best practices (Wild et al., 2022). Moreover, in software development education, incorporating real-world projects as motivational tools has been shown to bridge the gap between theory and practice, enhancing student engagement and skill acquisition (Dong et al., 2019), Collaborative projects, such as those addressing societal issues like homelessness, underscore the potential of faculty-student collaboration in fostering experiential learning and social responsibility (Snedker et al., 2023). Technological advancements further augment peer-to-peer teaching approaches, with multimedia learning principles playing a pivotal role in instructional design. Research emphasizes the integration of motivation into multimedia learning to enhance engagement and knowledge retention (Mayer, 2014), Additionally, leveraging commonly-available technologies enables the creation of online multimedia lessons, aligning with cognitive theories of learning (Cavanagh & Kiersch, 2023). Innovative tools, such as graphical coding environments and Python packages for geospatial analysis, provide interactive learning experiences and facilitate hands-on exploration of complex concepts (Mahadevan et al., 2016),Moreover, block chain-based incentive systems offer novel approaches to enhancing student participation in co-curricular activities, fostering a culture of collaboration and innovation (Mantry et al., 2023). Hackathons serve as platforms for open digital innovation, promoting collaboration and creativity among participants (Hew & Lo, 2018), Diverse initiatives aim to broaden participation in hackathons, fostering inclusivity and diversity within the technology community ((Longmeier et al., 2022); (Richard et al., 2015). Similarly, pair programming in educational settings facilitates collaborative learning experiences, promoting communication and problem-solving skills among students (Campe et al., 2020) .(F. Xu & Correia, 2023) conducted a systematic review and found that adopting distributed pair programming as a team learning activity can enhance learning outcomes in higher education settings. Their findings suggest that this approach fosters collaboration and problem-solving skills among students.(Richard et al., 2015) investigated the effects of gender composition within pairs on collaboration dynamics in a robotics workshop. They found that the gender composition of pairs influences collaboration behaviours, highlighting the importance of considering diversity in collaborative settings.(Robe & Kuttal, 2022) designed Pair

Buddy, a conversational agent for pair programming, aiming to enhance collaboration and productivity. Their study demonstrates the potential of conversational agents in facilitating effective communication and task coordination among programming pairs.(Sobral, 2020) evaluated the effectiveness of pair programming in higher education and discussed its potential as a learning strategy. Their findings suggest that pair programming promotes knowledge sharing, problem-solving skills, and student engagement.(W. Xu et al., 2023) conducted multimodal learning analytics of collaborative patterns during pair programming in higher education. Their study provides insights into the collaborative behaviours and communication patterns of programming pairs, facilitating a deeper understanding of pair programming dynamics.(Hawlitschek et al., 2023) conducted a literature review on empirical research related to pair programming in higher education. Their review synthesizes existing findings and identifies gaps in the literature, providing valuable insights for future research directions.

(Roque-Hernández et al., 2021) conducted a case study on acceptance and assessment in student pair-programming. Their findings suggest that pair programming enhances student engagement and learning outcomes, but effective assessment strategies are needed to evaluate individual contributions within pairs.(Collins, 2016) explored live coding and its role in teaching Supercollider, a programming language for audio synthesis and algorithmic composition. The study highlights the interactive and creative aspects of live coding sessions in facilitating music education.(Hofbauer et al., 2022) discussed teaching software engineering through programming over time, emphasizing the importance of project-based learning and iterative development processes in software engineering education.(Collins, 2016) investigated the use of Creative Commons sounds in live coding sessions. The study explores the potential of integrating sound-based creative practices into coding education, fostering interdisciplinary learning experiences.(Jamila, 2020) discussed teaching data science with literate programming tools, emphasizing the integration of coding and data analysis techniques in data science education.(Raj et al., 2018)examined the role of live coding in learning introductory programming. Their study highlights the interactive and experiential learning experiences facilitated by live coding sessions.(Dixit et al., 2021) demonstrated a live coding session with the cloud and a virtual agent, showcasing the potential of integrating cloud computing and virtual agents into live coding environments for educational purposes.

III. CODING METHODOLOGIES

A. Interactive online platforms

Interactive online platforms have revolutionized coding education, providing learners with dynamic and engaging environments to enhance their programming skills. These platforms offer interactive tutorials, coding challenges, and real-time feedback, making the learning process enjoyable and effective. Codecademy, Khan Academy, Udacity, edX, CodeCombat, CheckiO, HackerRank, GitHub, and Stack Overflow are prominent platforms fostering interactive and collaborative learning. These platforms promote hands-on coding practice, interactive problem-solving and real-world project development, creating a vibrant coding community.

1) Implementation:

Students use online resources such as GitHub, Stack Overflow, W3school, Google Colab, Kaggle, and ML Playground to solve problems including numerical integration, root equations, ordinary differential equations, statistical plotting, and exploratory data analysis. With the real-time feedback these platforms offer, students may practice coding right in their web browser and improve their problem-solving abilities. In order to promote a cooperative and encouraging coding environment, students also take part in group coding, make contributions to open-source projects, and have conversations.

2) Outcomes:

- Improved Collaborative Learning and Coding Proficiency: Students can practice coding in dynamic environments on interactive platforms, which helps them become more proficient in a variety of programming languages. Platforms like GitHub and Stack Overflow also encourage teamwork by letting students add to projects, post questions, and receive responses. This builds a helpful community that improves coding abilities and information exchange.
- Practical Project Experience and Interactive Problem-Solving: Students work on real-world projects where they apply their coding skills to real-world situations, preparing them for challenging tasks in the workplace. Coding assignments are made fun and interactive by interactive platforms like CodeCombat and HackerRank. These platforms support on-going learning and self-improvement by improving critical thinking and provide quick feedback.

3) Challenges:

- Overcoming Technical and Access Barriers: In order to provide every student with a flawless learning experience, technical obstacles related to platform compatibility and universal access to technology must be addressed.
- Personalized Assistance and Data Security: Ensuring the accuracy of interactive tutorials by implementing stringent quality control procedures and offering tailored support for varying learning speeds. Preserving students' enthusiasm and involvement in class while putting data security and privacy first on collaborative learning environments, protecting students' private information and academic integrity.

B. Live coding sessions

Teachers may now show students how to code in real time using live coding sessions, which have become a popular and dynamic teaching tool. Instructors engage students through problem-solving, software development, and live explanation of difficult algorithms through platforms such Google Colab, YouTube, and Udemy. With the help of this method, students can watch the coding process in action, ask questions, and acquire valuable insights that help them comprehend programming ideas on a deeper level.

1) Implementation:

Instructors use Python to solve issues and explain coding strategies in live coding sessions that take place on sites like Google Colab, YouTube, and Udemy. Using Python libraries, students investigate data visualization and participate in coding challenges during these sessions. Students can see the entire coding process in real-time during these sessions, which improves their ability to solve problems and program..

2) Outcomes:

- Rich Learning Experience: Students get a deep knowledge of programming topics through live coding sessions, where they watch seasoned programmers implement principles in real time. This improves their comprehension of complex algorithms and fosters a thorough awareness of coding approaches.

- Enhancement of Skills and Community Building: Students gain practical techniques to addressing coding problems and enhance their analytical and critical thinking abilities by receiving prompt feedback and following a methodical approach to problem-solving. Additionally, engaging in active coding exercises creates an interactive learning environment that enhances the overall learning experience and strengthens the bonds of community among students by promoting peer collaboration, inquiries, and conversations.

3) Challenges:

- Mitigating Technical Challenges: Preserving the entire learning experience for students requires resolving technical difficulties like platform bugs and internet connectivity challenges in order to maintain the ongoing flow of live coding sessions.
- Keeping the Content Up to Date and Relevant: It's critical to keep the content offered in live sessions current, relevant, and curriculum-aligned. Moreover, creative teaching strategies and interactive exercises are necessary to keep students' interest and active involvement at high levels, especially during longer sessions. To balance the length of a session, cover important topics, and guarantee student participation and attentiveness, effective time management tactics are required.

C. Pair Programming

Pair programming was introduced as a pedagogical tactic to teach data analytics, statistical methodologies, and numerical methods in a course on data science and numerical analysis. Enhancing students' problem-solving abilities, encouraging group projects, and strengthening their grasp of how to apply statistical and mathematical ideas to real-world data analysis were the main objectives.

1) Implementation:

Random pairings of students ensured a diversity of backgrounds and skill sets. Assignments pertaining to data analytics, statistical analysis, and numerical approaches were provided to each couple. When it came to developing the code, one student was the "driver," while the other was the "observer," going over the code, offering suggestions for improvement, and giving comments. Regular role rotation was used to guarantee equitable participation and learning. Students work in pairs during Google Colab programming sessions to design an engine data dashboard using Microsoft Excel. Students use the real-time collaboration tools offered by Google Colab to code and create the dashboard as a group. This exercise combines data visualization approaches with programming knowledge to improve their communication and teamwork abilities. By working together, the students create visually appealing and educational dashboards that demonstrate their ability to apply programming knowledge to real-world data analysis issues.

2) Outcomes:

- Development of Collaborative Skills: Pair programming promoted deep comprehension of statistical and numerical topics by enhancing collaborative problem-solving. Students gained quick feedback that improved their communication and problem-solving tools and gave them more confidence when using the methods on actual datasets.
- Effective Learning and Diverse Views: While collaborative coding introduced students to a variety of problem-solving techniques, the quick error feedback loop in pairs encouraged effective learning. This method increased their comprehension, provided a safe learning atmosphere, and gave them the confidence to take on challenging tasks.

3) Challenges:

- Ability Differences: Advanced students often needed to put in more work to explain concepts to their partners because pairs occasionally had different ability levels. Peer education and group discussions assisted in lessening this difficulty.
- Time management: Setting up the schedules for the pair programming sessions required careful planning, flexible scheduling, and efficient time management techniques.

D. Flipped Classroom Model

A flipped classroom approach was used to increase student engagement and promote learning results in a university course that focused on statistical approaches, data analytics, and numerical methods. In place of traditional lectures, pre-recorded video lectures, online tutorials, and reading assignments were used to help students become familiar with the theoretical concepts on their own. Interactive workshops replaced traditional lectures, emphasizing real-world applications, group problem-solving and in-depth conversations.

1) Implementation:

Under the flipped classroom paradigm, students watch YouTube videos and PowerPoint presentations on their own to learn about data preprocessing, handling outliers, and Python libraries (Pandas, NumPy, Scikit-Learn). They work through practical tasks in class, using guided instruction to apply the ideas they have learned to real datasets. This method encourages independent study, engaged engagement, and the development of transferable skills in data analysis and manipulation.

2) *Outcomes:*

- Encouraging Active Participation: The flipped model made sure students were ready for in-class workshops by encouraging them to participate actively. In order to close the gap between academic knowledge and practical application, these seminars promoted thoughtful debates and hands-on problem-solving exercises.
- Enhanced Critical Thinking and Collaborative Problem-Solving: Through case studies and group projects, students were prodded to work together to analyze datasets and interpret findings in order to solve problems more collaboratively. Understanding was enhanced as a result of the instructors' tailored help, which addressed each student's unique difficulties. Through the use of authentic datasets, students were able to refine their critical thinking skills, assess various strategies, select the best methods, and develop their analytical reasoning.

3) *Challenges:*

- It was difficult to guarantee that every student possessed the necessary technology skills to utilize online resources and take part in interactive workshops. This was related to technological literacy. If more assistance or training was required, it was given.
- Time management: It was important to carefully schedule the pre-class activities and in-class activities to allow students adequate time to individually assimilate the theoretical concepts.

E. Interactive Coding Environments

Interactive coding environments were added to a university program that teaches data analytics, statistical approaches, and numerical methods in order to improve student learning. The integration of CodeAcademy, Jupyter Notebooks and RStudio into the curriculum enables students to actively explore and comprehend complicated concepts by enabling them to conduct statistical analyses, view data, and experiment with algorithms in real-time.

Implementation: Jupyter Notebooks were taught to the students as interactive coding environments. The lectures included interactive exercises and live demos so that students could work with code snippets, see datasets, and see results right away. Students were encouraged to use these environments for data visualization, statistical modelling, and numerical simulation tasks through in-class assignments and group projects. Students investigate machine learning regression and classification case studies using engine data, as well as exploratory data analysis techniques, in interactive coding environments such as Jupyter Notebooks. Students obtain hands-on experience in data analysis, visualization, and machine learning applications by utilizing Jupyter Notebooks. This strategy improves their comprehension of real-world datasets and fortifies their competence in machine learning and exploratory data analysis techniques.

1) *Outcomes:*

- Interactive settings made it possible to explore statistical and numerical approaches hands-on and provided immediate feedback, which helped to bring abstract concepts to life. Instant feedback on their code allowed students to learn more quickly and felt more confident in their coding abilities.
- Data Visualization and Collaborative Learning: In order to effectively communicate statistical results, students were able to generate interactive charts and graphs thanks to interactive platforms that enabled dynamic data visualization. Through group projects, shared code, and feedback, collaborative problem-solving was promoted; strengthening shared learning opportunities and building a feeling of community. In order to gain a deeper knowledge of algorithms and statistical techniques, students could experiment in real-time, changing code and monitoring changes.

2) *Challenges:*

- Technical Proficiency: At first, several students found it difficult to use interactive coding environments due to the technical aspects. To assist students in mastering the use of these platforms, workshops and tutorials were held.
- Resource Availability: There was a problem making sure every student had access to the necessary hardware and software. To lessen this problem, computer labs were outfitted, and cloud-based solutions were investigated.

F. Multimedia Learning

In order to improve student comprehension and engagement in a course on data analytics, statistical approaches, and numerical methods, a multimedia learning strategy was used. To make abstract mathematical and statistical concepts more approachable and interesting, interactive films, podcasts, visualizations, and animations were incorporated into the curriculum to explain difficult ideas.

1) *Implementation:*

We produced interactive simulations, info graphics that are visually engaging, podcasts with expert interviews, and educational films. Lectures, online modules, and assignments all made use of these tools. Students' understanding of complex ideas was aided by the visual explanation of statistical models, data analytics approaches, and complex algorithms through animations and real-world applications. Students use animations and movies from YouTube to understand bearing fault diagnosis in multimedia learning settings. By offering both visual and audio assistance, these multimedia resources improve the educational process. Through interactive materials, students gain a thorough comprehension of bearing problem diagnosis by applying both theoretical knowledge and real-world applications. Students can better understand difficult subjects and develop their diagnostic skills with the help of this multimedia method.

2) *Outcomes:*

- Improved Learning Experience: Students' understanding and recall of mathematical and statistical concepts were improved by multimedia resources that appealed to a variety of learning styles. These materials included interactive visualizations and real-world examples.

- Engagement and Industry Relevance: Students were drawn in by interactive films and animations that turned abstract concepts into relatable information and inspired them to investigate mathematical and statistical techniques. Podcasts with business professionals provide insightful real-world information that helped students better understand the material by bridging the gap between academic theory and real-world applications.

3) Challenges:

- Creation of Resources: It took a lot of effort and knowledge to produce multimedia content of a high calibre. For the information to be accurate and successful, cooperation with educators and multimedia specialists was crucial.
- Technical Accessibility: Ensuring that all students had access to multimedia resources, particularly in places with poor internet connectivity, was difficult. By offering alternate formats and offline access, this problem was lessened.

G. Real-World Projects

A pedagogical move toward real-world projects was made in an advanced university course that focused on data analytics, statistical approaches, and numerical methods. As part of their projects, students had to use statistical and mathematical ideas to solve real-world issues, which promoted critical thinking and practical application.

1) Implementation:

Students work on real-world projects where they use a Tensile Test rig and Google Colab to identify mechanical properties through the use of stress-strain curves. Students have practical experience testing materials by using the Tensile Test rig for experiments and Google Colab for data processing. By extracting crucial mechanical properties from stress-strain curves, they improve their comprehension of how materials behave in various scenarios. Students gain important skills for real-world engineering applications and research through this hands-on approach.

2) Outcomes:

- Applied learning and Problem-Solving: By applying abstract ideas like machine learning algorithms and regression analysis to actual datasets, students were able to gain a deeper comprehension of intricate mathematical techniques. Their ability to adapt to problems and refine solutions through iteration improved their problem-solving skills and sharpened their analytical and critical thinking abilities.
- Real-world projects promoted collaborative learning by allowing students to take use of one another's abilities and promote teamwork and efficient task division. Collaborative projects also fostered industry engagement. By interacting with industry partners, they were able to gain useful insights, better align their abilities with market demands, and increase their employability. Student interaction with mentors enhanced their project management, communication, and presentation skills, all of which aided in their overall professional development.

3) Challenges:

- Project Scope: Defining projects that were challenging yet manageable within the course duration was a constant challenge. Careful selection and scoping of projects were necessary to ensure meaningful outcomes.
- Resource Allocation: Ensuring students had access to necessary resources, datasets, and tools required meticulous planning. Collaboration with external partners and institutions helped bridge resource gaps.

H. Peer-to-Peer Teaching

Peer-to-peer learning was implemented in an academic program that covered statistical approaches, data analytics, and numerical methods. Under the direction of the professors, students were encouraged to work together, share knowledge, and instruct one another. The objectives of this approach were to improve comprehension, promote participation, and create a welcoming learning environment.

1) Implementation:

Peer-to-peer teaching involves assigning students' specific topics to study on their own, with a concentration on data visualization with the Seaborn and Matplotlib libraries. These subjects become the subject of group discussions and presentations during the practical sessions. Students discover cutting-edge data visualization methods through collaborative learning, share knowledge, and collectively improve their abilities to produce meaningful visual representations. Through active participation, information exchange, and cooperative problem-solving, this method promotes a deeper comprehension of data visualization principles among peers.

2) Outcomes:

- Improved Knowledge and Intense Participation: Mentoring colleagues required a thorough mastery of the subject matter, encouraging thorough understanding and clear explanation. In order to create an interactive learning environment where questions lead to meaningful dialogues, presenters and listeners actively participated in discussions.
- Better Communication, a Helpful Community, and Confidence: Students' communication skills improved as a result of the issues being presented and they were better able to articulate difficult ideas. Peer-to-peer instruction promoted curiosity and teamwork in a nurturing learning environment. Students' confidence was increased when they presented in front of their classmates, which had a favourable effect on their overall academic performance, sense of camaraderie in the classroom, and presentation skills.

3) Challenges:

- It was difficult to make sure that the teaching materials catered to all of the students because of their varied backgrounds and competence levels. The content has to be carefully planned and modified.
- Effective time management was necessary to create a balance between the normal curriculum and peer-led sessions. For all subjects to be addressed in an efficient manner, planning and arranging were essential.

I. Storytelling and Scenarios

A storytelling and scenario-based teaching style was implemented in a university course on data analytics, statistical approaches, and numerical methods. Contextualizing difficult mathematical and statistical ideas was intended to help pupils relate to and enjoy learning about them. The curriculum was designed to incorporate real-world events and narratives that illustrate the practical applications of statistical and numerical approaches.

1) Implementation:
Through storytelling and scenario-based exercises on sites like Kaggle and Google Colab, students work on topics like classifying human pulse rates and predicting the energy use of appliances. Students use these real-world scenarios to put their knowledge to use on sites like Kaggle and Google Colab, where they develop predictive models for things like appliance energy usage trends and human pulse rate. Students improve their ability to solve problems by analyzing data trends and creating prediction algorithms, and they receive practical experience in data-driven decision-making. Through these projects, students can use data to generate engaging tales that connect theory to practical applications.

2) Outcomes:
- Enhancing Problem-Solving and Contextual grasp: Students gained a contextual grasp of abstract topics by applying statistical and numerical methodologies to real-world problems. As students took into account a variety of variables and uncertainties, authentic issues promoted the development of problem-solving abilities and analytical reasoning.
- Higher Engagement, Better Application Skills, and Holistic Learning: Students' attention was piqued and active engagement and discussion was encouraged by the use of storytelling techniques and real-life narratives. Student's practical application skills improved when they analysed real-world difficulties, which enabled them to modify their theoretical understanding to address difficult problems. These scenarios also offered a comprehensive perspective that covered the technological, moral, social, and economic ramifications, promoting a well-rounded education.

3) Challenges:
- Resource Integration: Careful selection and integration into the curriculum were necessary to include real-life events. It was difficult to find diverse and pertinent examples that appealed to the various interests and backgrounds of the students.
- Evaluation Techniques: It was difficult to create exams that assessed students' comprehension of theoretical ideas as well as how they applied them in real-world situations. It was essential to develop thorough and equitable evaluation procedures.

J. Hackathons and Coding Competitions

Hackathons and coding contests were introduced as hands-on learning opportunities in an academic program that focuses on numerical methods, statistical methodologies, and data analytics. These activities promoted students' application of knowledge and abilities in a cooperative and competitive setting, allowing them to solve practical issues and demonstrate their mastery of statistical and numerical techniques.

1) Implementation:
Students take part in short projects based on statistical metrics in hackathons and coding competitions organized by sites such as Kaggle and HackerRank. Through these contests, students can practice applying statistical ideas to address real-world problems in a collaborative and competitive context. Students learn about statistical measurements and their applications better when they work on practical projects. Students are better equipped to examine data, make insightful conclusions, and successfully communicate their findings thanks to these activities, which also promote creativity, teamwork, and problem-solving abilities. Attending hackathons and coding contests gives students the chance to demonstrate their skills and get useful experience using their statistical knowledge to solve real-world problems.

2) Outcomes:
- Practical Application of Numerical and Statistical Techniques: Hackathons and coding competitions gave participants hands-on experience and improve confidence in their ability to tackle real-world problems and use academic knowledge in an effective manner.
- Enhancing Teamwork and Time Management Skills: By working under pressure to replicate actual project deadlines, participants refined their time management techniques. In team-based competitions, collaborative learning promoted knowledge sharing, learning from peers, and utilizing a variety of skills, all of which improved participants' capacity for teamwork.
- Industry Professionals' Involvement: Students were exposed to genuine industry difficulties and received feedback, which helped to bridge the gap between academic understanding and professional applications. Judges' comments offered insightful feedback. Hackathons stimulated ingenuity and inventiveness by asking participants to come up with original solutions and push the limits of their expertise.

3) Challenges:
- Technical Difficulties: It was difficult to provide equal opportunities for players of different ability levels. It took significant planning to create tasks that would appeal to both novice and expert learners.
- Allocation of Resources: Major resources, such as a location, judges, mentors, and awards, were needed to organize hackathons and coding contests. It was critical to obtain funding and assistance from the academic institution and sponsors.

K. Incorporate IoT and Robotics

Robotics and the Internet of Things (IoT) were part of the curriculum of a multidisciplinary academic program that focused on data analytics, statistical approaches, and numerical procedures. Giving the students a solid understanding of these technologies and their possible uses in statistical and numerical analysis was the aim. The goal of this novel approach was to prepare students for the quickly developing field of data-driven decision-making in the context of robots and the Internet of Things.

1) Implementation:

Drone programming and Internet of Things applications are the main topics of professional lectures from the industry that are integrated into robotics and IoT courses. Students gain a grasp of how Internet of Things (IoT) technology can be combined with drones through the study of drone sensors, data collection, and real-time monitoring. In their exploration of drone programming, they also learn about drone code and the various uses for drones. With the guidance of experienced industry executives, students acquire a deeper understanding of state-of-the-art technology and are better prepared for professions in the constantly expanding fields of robotics and IoT. These courses function as a bridge between theory and practice to help students create and offer meaningful solutions in the robotics and Internet of Things fields.

2) Outcomes:

- Real-world Multidisciplinary Experience: Students learned how to collect and analyze data in real-time while becoming proficient with robotics platforms and IoT devices. An interdisciplinary viewpoint was made possible by the combination of IoT, robotics, and data analytics. This perspective highlighted the benefits of combining numerical approaches, statistical analysis, and real-world application while bridging the divide between hardware and software.
- Better Problem-Solving and Industry Readiness: Working with IoT and robotics posed difficult problems, honing students' problem-solving abilities. By fostering creativity and invention, these technologies helped students come up with innovative ideas that led to the development of useful applications like industrial automation and predictive maintenance. Students were better equipped for jobs in automation, the Internet of Things, and data-driven decision-making as a result of this exposure, which increased their industry relevance.

3) Challenges:

- The integration of IoT devices and robotics platforms necessitated significant resources, such as specialized equipment and knowledgeable instructors. This brings us to our second point of resource intensity. Getting technical know-how and financing was difficult.
- Diversification of Skills: Students' knowledge with robotics and IoT technologies varied widely. It took some juggling to create a program that would challenge students of all ability levels and also allow for diversity.

L. Online Coding Communities

(Dash et al., 2022) Online coding communities have been included into a curriculum to improve learning in an academic program that emphasizes data analytics, statistical methodologies, and numerical methods. To establish a cooperative and encouraging online learning environment for students, sites such as Stack Overflow, Kaggle, and GitHub were employed. Collaborative coding and data analysis initiatives provided these communities with a platform for learning, information exchange, and issue resolution in the real world.

1) Implementation:

Students work together on projects like centrifugal pump performance testing in online coding communities like GitHub, Stack Overflow, and Kaggle. Students use these platforms to share knowledge, collaborate on coding projects, and solve centrifugal pump testing-related problems. They are able to use pooled resources, consult with specialists, and help create novel testing techniques. By fostering a feeling of community through collaborative learning, students can take advantage of the collective expertise and obtain hands-on experience in centrifugal pump performance analysis and testing methodologies. Students that actively participate in online coding communities improve their problem-solving abilities and broaden their comprehension of practical engineering applications.

2) Outcomes:

- Peer Learning and Community Support: Students were able to ask questions, exchange answers, and gain knowledge from the experiences of their peers by using the platform that online coding communities provide for peer learning. This improved their educational experience by fostering a sense of community and friendship.
- Practical Use and Networking Possibilities: Students were able to apply theoretical knowledge to real-world problems through the participation in open-source projects and real-world coding challenges, which improved their problem-solving abilities. Being a part of online forums also made it easier for them to network with experts and professionals around the world, which widened their horizons and exposed them to various approaches and best practices in data analytics and numerical methods.

3) Challenges:

- Quality Control: It was difficult to be sure that data and solutions from internet forums were reliable. Teachers needed to mentor pupils in the critical assessment and use of web-based information.
- Time management: Managing assignments, participation in online communities, and other academic obligations required efficient time management abilities. Guidance on how to prioritize their online interactions was required for students.

M. Reflective Learning

Reflective learning was used in an academic curriculum that concentrated on statistical approaches, data analytics, and numerical methods. Self-awareness, critical thinking, and comprehension via introspection are highlighted in reflective learning. It was recommended of the students to think back on their experiences, difficulties, and methods used in statistical and numerical analysis. Journaling, group discussions, and self-evaluations were all incorporated into the curriculum on a regular basis.

1) Implementation:

Students are given tasks such as developing cheat sheets for Python libraries and summarizing scripts and grammar as part of reflective learning exercises. Students are encouraged to reflect on their learning through this project, which summarizes difficult programming ideas in a clear and succinct manner. Students strengthen their comprehension of Python libraries, improve their coding abilities, and hone their effective communication skills by making these cheat sheets. Through tasks that encourage reflection and self-evaluation, students can strengthen their knowledge and improve their ability to communicate complicated technical material in an understandable and concise manner.

2) Outcomes:

- Deeper Understanding and Metacognition: Students were encouraged to introspect through reflective learning, which improved their comprehension of mathematical and statistical topics and provided them with insights into their own mental processes. Through the development of metacognitive skills, they became aware of their learning tactics, strengths, and shortcomings, which made it possible to adjust their ways more effectively.
- Better Communication and Problem-Solving: Through reflective exercises, problem-solving strategies were critically analyzed, procedures were improved, and problem-solving abilities were strengthened. Students' ability to express ideas clearly, defend positions, and offer helpful criticism improved through group discussions, which promoted efficient communication in technical talks. Instilling a sense of ownership through reflective learning also encouraged pupils to overcome obstacles and develop a positive attitude toward learning.

3) Challenges:

- Subjectivity: Because reflections are subjective by nature, it is difficult to assess them impartially. To guarantee uniformity in evaluation, structured norms and rubrics were created.
- Time Intensity: Adding introspective exercises required more time in class. Careful planning was needed to maintain a thorough learning experience while balancing these activities with the core curriculum.

N. Activities/ Tools/ Platforms used

These strategies are utilised for course Numerical and Statistical Methods (NSM) and Artificial Intelligence and Machine Learning (AIML) of Third Year Mechanical Engineering along with Data Analytics Lab (DAL) of Final year Mechanical Engineering of Savitribai Phule Pune University.

Table 1 shows the Topic wise Activities/ Tools/Platforms Used.

TABLE I
TOPIC WISE ACTIVITIES/ TOOLS/PLATFORMS USED

Methodology	Activities/ Tools/ Platforms used	Topics/ Experiments Covered	Time Allocated	Subject - NSM	Subject - DAL	Subject- AIML
Interactive Online Platforms	GitHub, Stack Overflow, W3school, Google Colab, Kaggle, ML Playground	Root of the Equation, Numerical Integration, Ordinary differential Equations, Statistical Plots, Exploratory Data Analysis	1 Practical - 2hours	Assignment 1,3,5,7	Assignment 1	Assignment 1
Live Coding Sessions	Google Colab, Video Tutorial available on YouTube, Udemy	Problem Solving with Python, Data Visualization with Python Framing	1 Practical - 2hours		Assignment 2	Assignment 1,2
Pair Programming	Google Colab	Dashboard for Engine Data using Microsoft Excel	1 Practical - 2hours		Assignment 3	Assignment 3
Hackathons and Coding Competitions	Kaggle, HackerRank	Mini Project based on Statistical Measures	1 Practical - 2hours	Assignment 7		
Flipped Classroom Model	YouTube,PowerPoint Presentations	Data Pre-processing, Outliers Handling, Pandas, Numpy, Sklearn Python Libraries	1 Practical - 2hours	Assignment 2	Assignment 1,2,4	Assignment 9

Interactive Coding Environments	Codecademy	Exploratory Data Analysis, Machine Learning Regression and Classification case studies on Engine Data	1 Practical - 2hours		Assignment 1,5	Assignment 8
Multimedia Learning	YouTube Videos, Animations	Bearing Fault Diagnosis	1 Practical - 2hours		Assignment 6	Assignment 7
Real-World Projects	Google Colab, Tensile Test rig	Finding Mechanical Properties using Stress Strain Curve	1 Practical - 2hours		Assignment 2	Assignment 8
Peer-to-Peer Teaching	Few topics were given for self study, and same topics were taken for group discussion and presentation during practical sessions	Data Visualization using seaborn, Matplotlib library	1 Practical - 2hours	Assignment 4	Assignment 7	Assignment 4,6
Storytelling and Scenarios	Kaggle, Google colab	Human Heart rate Classification, Appliances energy Prediction	1 Practical - 2hours		Assignment 8	
Incorporate IoT and Robotics	Expert Lecture by Industry Person	IoT in Drone, Drone Programming	2 Practical - 4hours		Students workshop on Drone Programming	
Online Coding Communities	GitHub, Stack Overflow, and Kaggle	Centrifugal Pump Performance Testing	1 Practical - 2hours		Assignment 9	Assignment 5
Reflective Learning	Assignment	An Assignment was given to prepare cheat sheet for python libraries to summarise the codes and syntax	Each Practical - 15 minutes	All Assignments	All Assignment	All Assignment

IV. RESULTS AND DISCUSSION

To validate outcomes and satisfaction of students, the feedback of 73 Students was taken on Google form, students were asked to rate their experience with these coding methodology on scale of 1 to 10(Low to High). Feedback analysis is done with python coding to explain the outcomes. Fig. 4 shows the boxplot of rating of methodology. Mostly students like almost every methodology on an average scale of 7 to 9. Figure 1 shows the Boxplot of Ranting for Methodologies.

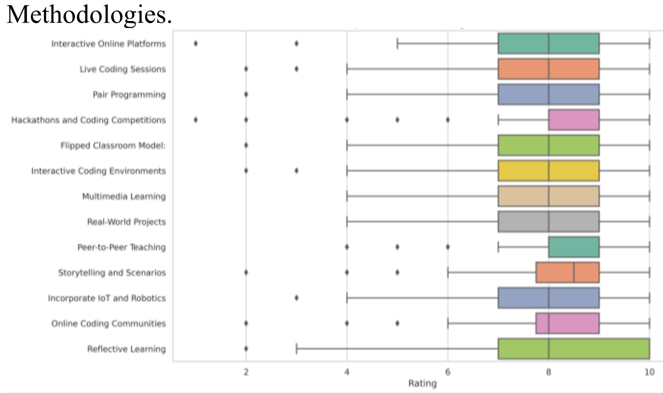

Fig. 1. Boxplot of Ranting for Methodologies

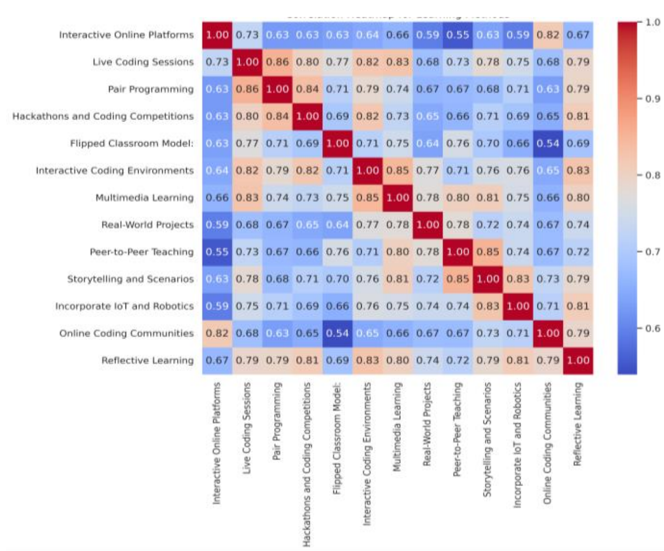

Fig. 2. Correlation of all methodologies with each other

Figure 2 shows the Correlation of all methodologies with each other. Second highest correlation (0.86) is between methods: [('Live Coding Sessions' and 'Pair Programming')]
Second lowest correlation (0.54) is between methods: [('Flipped Classroom Model' and 'Online Coding Communities')]

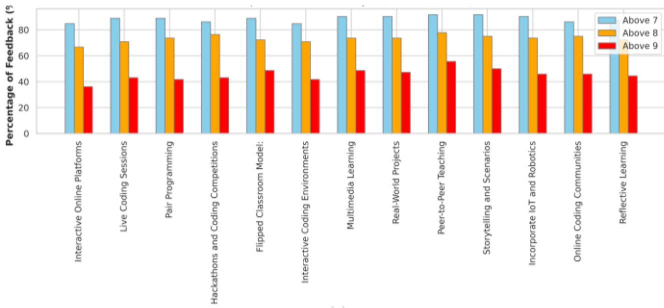

Fig 3: Feedback rating for 70 to 100% satisfaction

Figure 3 shows the Feedback rating for 70 to 100% satisfaction
It is observed that
Average percentage of feedback >= 7 is 88.46
Average percentage of feedback >= 8 is 73.18
Average percentage of feedback >= 9 is 45.51
Hence target (70% Satisfaction) of indirect assessment is achieved.
95.36% students are interested in active learning strategies.
Figure 4 shows the Category wise satisfaction of students.

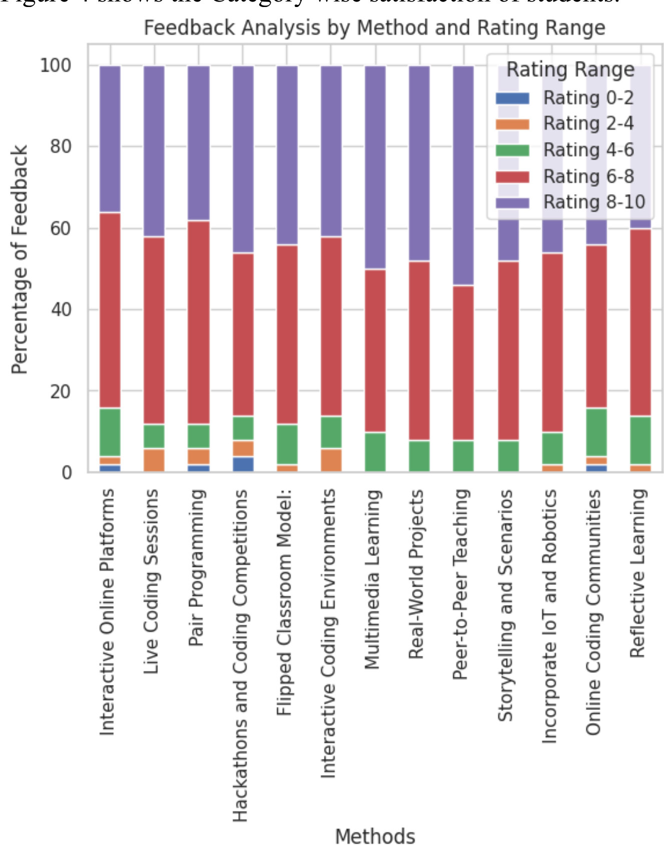

Fig. 4. Category wise satisfaction of students

Rating 0-2 is considered as Disagree, Rating 2-4 is considered as Neutral, Rating 4-6 is considered as agree, Rating 6-8 is considered as Moderately agree, Rating 8-10 is considered as Strongly Agree. As per feedback analysis almost 88% students are strongly agree for effectiveness of using these methodologies. Benefits for Slow Learners and Advanced learners as shown in table 2.

TABLE II
BENEFITS FOR SLOW LEARNERS AND ADVANCED LEARNERS

Strategy	Benefits for Slow Learners	Benefits for Fast Learners
Interactive Online Platforms	Access resources at their own pace, revisit difficult concepts	Explore advanced topics, participate in discussions, take on additional challenges
Live Coding Sessions	Real-time demonstrations, immediate feedback, ask questions	Engage in complex problems, contribute to sessions, help peers
Pair Programming	Support and guidance from a partner	Reinforce knowledge by teaching, tackle challenging parts of code
Hackathons and Coding Competitions	Develop skills through collaboration, real-world problem-solving	Push limits with complex problems, innovate solutions
Flipped Classroom Model	Review materials at own pace, prepare questions for class	Grasp basics quickly, use class time for deeper exploration
Interactive Coding Environments	Experiment with code, receive instant feedback	Explore advanced features, create complex projects
Multimedia Learning	Engage with videos, animations, simulations	Explore supplementary resources, gain deeper insights
Real-World Projects	Practical application of concepts, relevant learning	Tackle complex aspects, explore innovative solutions, lead project teams
Peer-to-Peer Teaching	Personalized explanations, support from peers	Reinforce learning by teaching, delve deeper to answer questions
Storytelling and Scenarios	Understand concepts through relatable stories	Create scenarios, apply concepts creatively
Incorporate IoT and Robotics	Hands-on, tangible learning	Explore complexities, create sophisticated projects
Online Coding Communities	Access support, ask questions, find resources	Engage in advanced discussions, share knowledge, contribute to projects
Reflective Learning	Reflect on progress, identify improvement areas	Reflect on successes, set goals, deepen understanding of advanced topics

A. *Connection with Existing Literature:*

Our study aligns closely with existing literature on educational methodologies and their impact on learning outcomes. Similar to prior research, we find that integrating diverse methodologies such as reflective learning, peer-to-peer teaching, and real-world projects enhances student engagement and facilitates deeper understanding across disciplines (Díaz-Iso et al., 2019; Weng et al., 2022; Dong et al., 2019). Our results corroborate the positive effects reported in literature, emphasizing critical thinking, collaborative learning environments, and practical skill acquisition (Kaczkó & Ostendorf, 2023; Pashnik et al., 2017; Mayer, 2014). While our study reaffirms these findings, it also contributes by focusing specifically on their application in the context of teaching coding skills to mechanical engineering students, thereby bridging a gap in the literature regarding technical education.

Implications: Our study suggests that integrating diverse teaching methodologies in engineering education can lead to enhanced student engagement, practical skill development, and interdisciplinary learning. By embracing new technologies and fostering collaboration, engineering programs can better prepare students for dynamic career challenges and promote long-term innovation in the field.

Limitations of the Study: Every study has its limitations, and ours is no exception. Some potential limitations include the challenge of assessing long-term retention of coding skills beyond the study duration, the variability in student backgrounds impacting outcomes, and the reliance on self-reported data for measuring learning outcomes immediately post-course. These limitations suggest avenues for further research to explore the sustainability of skill retention and address potential biases in data collection methods.

Future Scope of the Study: Our study opens avenues for future research in several directions. Firstly, exploring advanced applications of coding skills in mechanical engineering, such as machine learning algorithms for predictive maintenance or optimization tasks, could deepen the integration of coding within the discipline (Rahman et al., 2022). Additionally, investigating the scalability of our teaching methodologies across different educational settings and evaluating their impact on interdisciplinary collaborations could provide broader insights into optimizing technical education. Moreover, studying the effectiveness of integrating emerging technologies like IoT and blockchain in enhancing coding education could further enrich the scope of our findings (Hew & Lo, 2018; Mantry et al., 2023).

Interactive online platforms like GitHub, Stack Overflow, and Kaggle promote collaborative learning on real-world problems. Live coding sessions on platforms such as Google Colab offer hands-on experience in Python and data visualization. Pair programming and participation in hackathons enhance teamwork, creativity, and problem-solving skills. Flipped classrooms, interactive coding environments like Jupyter Notebooks, and multimedia resources make learning practical and engaging. Real-world projects, peer-to-peer teaching, storytelling in technical projects, and involvement in IoT and robotics broaden students' horizons. Online coding communities and reflective learning assignments further support continuous improvement and skill development.

These methods collectively prepare students for dynamic careers by equipping them with essential skills and practical experience in tech-driven environments.

Declaration: Author own this study and the content created by the Authors own Copyright AI tool and have proofread all content for correctness and accuracy.

CONCLUSION

The educational framework outlined here encompasses a rich tapestry of teaching methodologies, interactive tools, and real-world applications. Through engaging platforms like GitHub, Stack Overflow, and Kaggle, students collaborate on authentic challenges, refining problem-solving skills. Live coding sessions and multimedia learning deepen technical understanding, while pair programming and peer-to-peer teaching foster teamwork and knowledge exchange. Hackathons, competitions, and storytelling projects inject creativity into learning, encouraging innovative thinking. Integration of IoT, robotics, and expert lectures provides exposure to cutting-edge technologies and industry insights. Real-world projects bridge theory and practice, equipping students with practical skills for professional endeavours. Reflective learning activities promote self-assessment and solidify core concepts, ensuring a strong foundation. This comprehensive approach not only hones technical proficiency but also cultivates essential soft skills such as critical thinking, collaboration, and adaptability. In essence, this multifaceted educational strategy prepares students for a future where versatility and innovation are paramount. By combining theoretical knowledge with practical applications and fostering a holistic skill set, educators empower learners to excel in an ever-changing world, driving progress in Technology and data-driven field. The combination of interactive online platforms, live coding sessions, collaborative activities, real-world projects, and innovative teaching techniques creates a dynamic learning environment. These methods not only enhance technical skills but also promote teamwork, creativity, and problem-solving abilities. By integrating theory with practical applications and industry insights, educators prepare students for success in diverse fields, fostering adaptability and innovation. This multifaceted approach ensures a comprehensive education, empowering students to excel in the rapidly evolving landscape of technology and data-driven disciplines.

REFERENCES

Afzal, A., Munir, H., Khan, E. A., Ali, A., & Multan, B. (2021). Problems Faced by Elementary School Teachers (Est's) towards Slow Learners-Palarch's. In *Journal of Archaeology Of Egypt/Egyptology* (Vol. 18, Issue 2).

Arulkumar, K., Vaigundamoorthy, M., Prabaharan, N., & Subramaniam, U. (2022). Impact of the flipped classroom approach in engineering education: A course analysis. *Journal of Engineering Education Transformations*, *35*(4). https://doi.org/10.16920/jeet/2022/v35i4/22101

Bhoga, U., G, V. M., & A, D. B. (2024). Coding Club's Impact on Engineering Education. *Journal of Engineering Education Transformations*, *37*(IS2), 962–967. https://doi.org/10.16920/jeet/2024/v37is2/24145

Cacciamani, S., Perrucci, V., & Khanlari, A. (2018). Conversational functions for knowledge building communities: A coding scheme for online interactions. *Educational Technology Research and Development*, *66*(6). https://doi.org/10.1007/s11423-018-9621-y

Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education*, *30*(1). https://doi.org/10.1080/08993408.2019.1648119

Carbonell, J., Sánchez-Esguevillas, A., & Carro, B. (2017). From data analysis to storytelling in scenario building. A semiotic approach to purpose-dependent writing of stories. *Futures*, *88*. https://doi.org/10.1016/j.futures.2017.03.002

Cavanagh, T. M., & Kiersch, C. (2023). Using commonly-available technologies to create online multimedia lessons through the application of the Cognitive Theory of Multimedia Learning. *Educational Technology Research and Development*, *71*(3). https://doi.org/10.1007/s11423-022-10181-1

Collins, N. (2016). Live coding and teaching SuperCollider. *Journal of Music, Technology and Education*, *9*(1). https://doi.org/10.1386/jmte.9.1.5_1

Dahlgaard, J. J., & Dahlgaard-Park, S. M. (2006). Lean production, six sigma quality, TQM and company culture. *TQM Magazine*, *18*(3). https://doi.org/10.1108/09544780610659998

Díaz-Iso, A., Eizaguirre, A., & García-Olalla, A. (2019). Extracurricular activities in higher education and the promotion of reflective learning for sustainability. *Sustainability (Switzerland)*, *11*(17). https://doi.org/10.3390/su11174521

Dixit, B., Bedekar, M., Jahagirdar, A., & Sathe, N. (2021). Role of Active Learning Techniques in Development of Problem Solving Skills. *Journal of Engineering Education Transformations*, *34*(0), 670–674. https://doi.org/10.16920/jeet/2021/v34i0/157241

Dong, Y., Marwan, S., Cateté, V., Price, T., & Barnes, T. (2019). Defining tinkering behavior in open-ended block-based programming assignments. *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3287324.3287437

Dowsett, R. M., Green, M. S., & Harty, C. F. (2022). Speculation beyond technology: Building scenarios through storytelling. *Buildings and Cities*, *3*(1). https://doi.org/10.5334/bc.213

Ghosh, S., Agrawal, A., Ramu, B., Kumar, S. K., & Reddy, N. T. (2020). Application of C in Data Analytics. *Journal of Engineering Education Transformations*, *33*(0), 600. https://doi.org/10.16920/jeet/2020/v33i0/150128

Goy, R. W. L., Lee, J. S. E., Ithnin, F., & Sng, B. L. (2022). Current state of reflective learning in anaesthesiology residents: A call for “reflection-on-action” training in Singapore. *Singapore Medical Journal*, *63*(6). https://doi.org/10.11622/smedj.2020167

Hawlitschek, A., Berndt, S., & Schulz, S. (2023). Empirical research on pair programming in higher education: A literature review. In *Computer Science Education* (Vol. 33, Issue 3). https://doi.org/10.1080/08993408.2022.2039504

Hew, K. F., & Lo, C. K. (2018). Flipped classroom improves student learning in health professions education: A meta-analysis. *BMC Medical Education*, *18*(1). https://doi.org/10.1186/s12909-018-1144-z

Hofbauer, M., Bachhuber, C., Kuhn, C., & Steinbach, E. (2022). Teaching Software Engineering As Programming over Time. *Proceedings - 4th International Workshop on Software Engineering Education for the Next Generation, SEENG 2022*. https://doi.org/10.1145/3528231.3528353

Huang, H. Y., Lin, Y. J., Chen, Y. S., & Lu, H. Y. (2012). Imbalanced data classification using random subspace method and SMOTE. *6th International Conference on Soft Computing and Intelligent Systems, and 13th International Symposium on Advanced Intelligence Systems, SCIS/ISIS 2012*. https://doi.org/10.1109/SCIS-ISIS.2012.6505155

Jamila, E. H. (2020). Implementing a flipped classroom structure in engineering education to improve the soft skills. *Journal of Engineering Education Transformations*, *33*(3). https://doi.org/10.16920/jeet/2020/v33i3/147042

Kaczkó, É., & Ostendorf, A. (2023). Critical thinking in the community of inquiry framework: An analysis of the theoretical model and cognitive presence coding schemes. *Computers and Education*, *193*. https://doi.org/10.1016/j.compedu.2022.104662

Kimbrough, E. O., McGee, A. D., & Shigeoka, H. (2022). How Do Peers Impact Learning?: (Inline image) An Experimental Investigation of Peer-to-Peer Teaching and Ability Tracking. *Journal of Human Resources*, *57*(1). https://doi.org/10.3368/jhr.57.1.0918-9770R2

Longmeier, M. M., Dotson, D. S., & Armstrong, J. N. (2022). Fostering a Tech Culture through Campus Collaborations: A Case Study of a Hackathon and Library Partnership. *Science and Technology Libraries*, *41*(2). https://doi.org/10.1080/0194262X.2021.1963388

Mahadevan, A., Freeman, J., & Magerko, B. (2016). An interactive, graphical coding environment for EarSketch online using Blockly and Web Audio API. *Proceedings of the International Web Audio Conference*.

Mantry, P., Dhananjay, S., Desai, M., Kazi, S., & Patel, D. (2023). Blockchain-Based Incentive System for Enhancing Student’s Participation in Co-Curricular and Extracurricular Activities. *Lecture Notes in Networks and Systems*, *516*. https://doi.org/10.1007/978-981-19-5221-0_42

Mayer, R. E. (2014). Incorporating motivation into multimedia learning. *Learning and Instruction*, *29*. https://doi.org/10.1016/j.learninstruc.2013.04.003

Micle, D. E., Deiac, F., Olar, A., Drența, R. F., Florean, C., Coman, I. G., & Arion, F. H. (2021). Research on innovative business plan. Smart cattle farming using artificial intelligent robotic process automation.

Agriculture (Switzerland), *11*(5). https://doi.org/10.3390/agriculture11050430

Mudenagudi, U., Patil, U., Budihal, S., Tabib, R. A., Shruti, M., Satish, C., Iyer, N., & Shettar, A. (2019). A step towards introducing data analytics and visualization for students of electrical sciences: An initiative through machine learning course. *Journal of Engineering Education Transformations*, *33*(2).

Naicker, K., & Rensburg, G. H. van. (2018). Facilitation of reflective learning in nursing: Reflective teaching practices of educators. *Africa Journal of Nursing and Midwifery*, *20*(2). https://doi.org/10.25159/2520-5293/3386

Pashnik, B., Creta, A., & Alberti, L. (2017). Effectiveness of a Nurse-Led Initiative, Peer-to-Peer Teaching, on Organizational CAUTI Rates and Related Costs. *Journal of Nursing Care Quality*, *32*(4). https://doi.org/10.1097/NCQ.0000000000000249

Rahman, M. M., Khatun, F., Sami, S. I., & Uzzaman, A. (2022). The evolving roles and impacts of 5G enabled technologies in healthcare: The world epidemic COVID-19 issues. In *Array* (Vol. 14). https://doi.org/10.1016/j.array.2022.100178

Raj, A. G. S., Patel, J. M., Halverson, R., & Halverson, E. R. (2018). Role of live-coding in learning introductory programming. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3279720.3279725

Richard, G. T., Kafai, Y. B., Adleberg, B., & Telhan, O. (2015). StitchFest: Diversifying a college Hackathon to Broaden participation and perceptions in computing. *SIGCSE 2015 - Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/2676723.2677310

Robe, P., & Kuttal, S. K. (2022). Designing PairBuddy-A Conversational Agent for Pair Programming. *ACM Transactions on Computer-Human Interaction*, *29*(4). https://doi.org/10.1145/3498326

Roque-Hernández, R. V., Guerra-Moya, S. A., & Caballero-Rico, F. C. (2021). Acceptance and Assessment in Student Pair-Programming: A Case Study. *International Journal of Emerging Technologies in Learning*, *16*(9). https://doi.org/10.3991/ijet.v16i09.18693

Sharmin, S., Zingaro, D., Zhang, L., & Brett, C. (2019). Impact of Open-Ended Assignments on Student Self-Efficacy in CS1. *CompEd 2019 - Proceedings of the ACM Conference on Global Computing Education*. https://doi.org/10.1145/3300115.3309532

Snedker, K. A., Fredriks, A., & Nye, E. (2023). Counting Tents: Pedagogical Reflections on Faculty–Student Collaboration in a Real-World Project on Homelessness. *Teaching Sociology*, *51*(4). https://doi.org/10.1177/0092055X221134125

Sobral, S. R. (2020). Is pair programing in higher education a good strategy? *International Journal of Information and Education Technology*, *10*(12). https://doi.org/10.18178/ijiet.2020.10.12.1478

Stigmar, M. (2016). Peer-to-peer Teaching in Higher Education: A Critical Literature Review. *Mentoring and Tutoring: Partnership in Learning*, *24*(2). https://doi.org/10.1080/13611267.2016.1178963

Weng, C., Puspitasari, D., Rathinasabapathi, A., & Kuo, A. (2022). Reflective learning as an important key to the success of an online course. *Behaviour and Information Technology*, *41*(15). https://doi.org/10.1080/0144929X.2021.1988145

Wild, J., Siegrist, B., Hobohm, L., Münzel, T., Schwanz, T., & Sagoschen, I. (2022). Short and Concise Peer-to-Peer Teaching—Example of a Successful Antibiotic Stewardship Intervention to Increase Iv to Po Conversion. *Antibiotics*, *11*(3). https://doi.org/10.3390/antibiotics11030402

Xu, F., & Correia, A. P. (2023). Adopting distributed pair programming as an effective team learning activity: A systematic review. *Journal of Computing in Higher Education*. https://doi.org/10.1007/s12528-023-09356-3

Xu, W., Wu, Y., & Ouyang, F. (2023). Multimodal learning analytics of collaborative patterns during pair programming in higher education. *International Journal of Educational Technology in Higher Education*, *20*(1). https://doi.org/10.1186/s41239-022-00377-z

Yalagi, P. S., Dixit, R. K., & Nirgude, M. A. (2020). Enhanced Programming Learning Model (EPLM) through Continuous Collaborative Coding (CCC) Practice. *Journal of Engineering Education Transformations*, *33*(0), 561. https://doi.org/10.16920/jeet/2020/v33i0/150117