2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #52476

WIP: Interweaving Sustainability Education in Electrical and Computer Engineering Curricula

Prof. Amr Hassan, University of Pittsburgh

Dr. Amr Hassan (also know as Amr Mahmoud) received his B.Sc. degree in Electronics and Electrical Communications Engineering and the M.Sc degree in Engineering Physics from Cairo University, Egypt, in 2011 and 2015, respectively. He earned his PhD in Computer Engineering from the Electrical and Computer Engineering Department at the University of Pittsburgh, USA. Currently, he is an Assistant Professor in the ECE department at the University of Pittsburgh, and his expertise include designing neuromorphic computing systems, machine learning, especially deep learning for image processing and video prediction, and designing application-specific accelerators on FPGAs. Dr. Mahmoud is also dedicated to the engineering education development and innovation.

Dr. Irene B. Mena, University of Pittsburgh

Irene B. Mena has a B.S. and M.S. in industrial engineering, and a Ph.D. in engineering education. She has experience teaching programming, design, entrepreneurship, and sustainability topics, and is the Director of the First-Year Engineering Program at the University of Pittsburgh.

Dr. Mohamed A. S. Zaghloul, University of Pittsburgh

Mohamed A. S. Zaghloul was born in Cairo, Egypt, in 1987. He received his B.Sc. degree in Electronics and Electrical Communications Engineering in 2009, and his M.Sc. degree in Engineering Physics in 2012, both from the Faculty of Engineering at Cairo University. In 2019, he received his PhD from the Department of Electrical and Computer Engineering at the University of Pittsburgh, where he is currently working as an Associate Professor. His research interests are engineering education, and in the area of optics; he specializes in developing optical fiber-based sensors for monitoring harsh environments.

WIP: Interweaving Sustainability Education in Electrical and Computer Engineering Curricula

Abstract
The growing environmental impact of data-intensive computing highlights the need to embed sustainability within engineering education. This work-in-progress presents a pedagogical approach that integrates green computing and sustainable software development into core Electrical and Computer Engineering courses. Guided by principles of sustainable development, the project introduces modular learning units that connect software efficiency, hardware utilization, and computational decision-making to environmental and societal impacts. The modules leverage contemporary case studies, such as data center energy optimization and carbon-aware software practices, to reinforce sustainability as a technical competency rather than a peripheral topic. The pilot study targets junior- and senior-level students, impacting approximately 75 learners annually. A mixed-methods assessment framework is employed, combining performance-based evaluation with reflective surveys to measure both learning outcomes and shifts in student eco-consciousness. The overarching goal is to prepare future engineers to design environmentally responsible computing systems.

1. Introduction

The accelerating pace of artificial intelligence (AI) and data-intensive computing has its own toll on the environment. Training and operating large language models (LLMs) now requires enormous computational resources, as recent studies have shown. The training phase of a single AI model can generate over 625K pounds of CO2 gas—equivalent to almost five times the total emissions produced by an average American car over its lifetime [1]! As AI adoption continues to scale, data centers already account for ~4.4 % of U.S. electricity use (in 2023), and projections indicate that share could triple by 2028 [2], which is also the energy equivalent of a country like Argentina or Netherlands [3]. These realities reveal an urgent need to prepare the next generation of engineers to design, deploy, and manage computing systems through a lens of environmental responsibility.

Green and sustainable software engineering is an emerging discipline focused on the design and deployment of energy-efficient software and hardware systems that minimize environmental impact and reduce overall carbon footprint. The growing importance of this area is evidenced by initiatives such as the semiannual ranking of the world's top 500 green supercomputers, which promotes the adoption of energy-efficient high-performance computing systems [4]. Over the past decade, research in this domain has expanded rapidly to include advances in Green Information Technology (Green IT) [5, 6], the formulation of quantitative metrics for assessing software and system sustainability [7, 8], the development of sustainable software development practices [9, 10], and the design of energy-aware data structures and software libraries [11]. More broadly, green computing refers to the practices involved in the design, use, and end-of-life management of computing resources in an environmentally responsible manner while maintaining required computational functionality [12]. These practices include reducing the use of hazardous materials, optimizing energy efficiency across the product lifecycle, and promoting the reuse, recyclability, or biodegradability of computing components to mitigate environmental impact [12].

Despite the growing relevance of green and sustainable computing, computer engineering curricula have been relatively slow to integrate education in this area, as reflected by the limited number of academic publications addressing this topic [13]. As a result, several open research questions remain; and the authors would like to focus on the following three Research Questions (RQs): *1) How can sustainability-oriented modules be interwoven into deeply technical elective courses without displacing core content*, *2) Which evaluation strategies effectively measure changes in students' technical understanding, eco-consciousness, and future intentions*, and *3) In what ways did students' perceptions and knowledge of green computing and sustainable software development change as a result of the learning modules?*

In this work-in-progress, the authors explore a pedagogical approach that integrates green computing and sustainable software development concepts into core Electrical and Computer Engineering (ECE) courses, namely Algorithms for Big Data and Senior Design, under the engineering school hosting this study. The project aims to move sustainability from a peripheral ethical topic to a core technical competency by developing a set of modular learning units that explicitly connect green computing, hardware utilization, and software efficiency to their environmental and societal impacts. Students are exposed to contemporary case studies—from data center cooling optimization to carbon-aware software scheduling—highlighting how computational choices translate directly into energy use.

The remainder of this paper is structured as follows. Section 2 reviews relevant prior work in the area. Section 3 describes the courses examined in this study and outlines the design and implementation of the newly developed educational modules. Section 4 presents the assessment methods employed and discusses the resulting findings, and Section 5 concludes the paper.

2. Background and Previous Work

Several studies have highlighted the limited integration of green computing and sustainable software development education within higher education. In 2021, Saraiva et al. conducted a study involving 21 established researchers and educators in the field [13]. Their findings pointed to a significant shortage of dedicated courses and instructional materials addressing green computing, particularly within computer science curricula. The study further identified key challenges contributing to this gap, including limited expertise and resource constraints. While the authors noted that stand-alone courses in green computing or sustainable software development would be ideal, they concluded that embedding smaller, targeted educational modules across related courses is a more practical and scalable approach [13].

Complementing these findings, Damiano et al. surveyed 33 academic faculty members regarding the presence of educational modules related to green computing and sustainable software development in higher education [14]. The results revealed that such educational efforts remain underdeveloped due to multiple factors, including insufficient teaching materials, limited awareness of sustainability issues, and a lack of supporting technologies and tools [14].

More recently, Peters et al. analyzed 89 publications related to sustainability in computer engineering education [5]. Their study synthesized existing efforts from four perspectives: pedagogical approaches, covered topics, instructional building blocks, and learning outcomes. The analysis demonstrated that exposure to green computing education led to increased student awareness of the environmental impacts of the Information and Communication Technology (ICT) sector, shifts in student mindset, and heightened motivation to engage with sustainability challenges for broader societal benefit [5]. The authors emphasized the need for coordinated efforts among engineering educators to promote systemic and infrastructural changes within computer engineering curricula to support these emerging topics.

Despite these educational insights, evidence from industry-facing studies suggests that practicing software engineers continue to lack adequate training in sustainability. Pang et al. surveyed software developers via Reddit in 2016 and found that respondents demonstrated minimal understanding of energy efficiency concepts, limited knowledge of strategies for reducing software energy consumption, and uncertainty regarding how software energy usage is measured [15, 16]. These findings underscored an urgent need for structured education and training in this area. Similar concerns were also raised by Pinto et al., who observed a rapid increase in software energy consumption–related questions on Stack Overflow over the past decade, with responses often reflecting poor foundational knowledge of the field [17].

Consequently, the authors of this work were strongly motivated to initiate the sustainability education effort described in the following section, with the goal of encouraging similar initiatives across academic institutions, both within the United States and internationally.

3. Description of the Courses Used in this Study

The study was conducted within two courses, titled Algorithms for Big Data (ABD) and Senior Design (SD). The former is structured around two primary components. The first component focuses on the theoretical foundations of algorithms designed for the analysis and processing of large-scale datasets. In the second component, students apply these concepts through the implementation of four hands-on laboratory exercises based on Microsoft Azure® cloud computing services. These laboratories require students to construct data processing pipelines, optimize their performance, and analyze large datasets. Given the substantial computational demands and cloud resources required by the course, it provides a natural and relevant context for introducing concepts related to green computing and sustainable software development, with an emphasis on energy-efficient and environmentally responsible implementations.

The SD course is centered around a capstone project. Senior students enroll in this course and work on a semester-long project, where they get to propose an idea based on a real-life problem/need, design a solution that involves both hardware and software components, and make a working prototype. Throughout the semester, the instructors lecture the students on different important aspects of their design, how to optimize their design, and how to mitigate different challenges and problems they may encounter. Students have different milestones to submit, with the two most important ones being the midterm and the final reports.

3.1. Green Computing and Sustainable Software Development Modules Developed

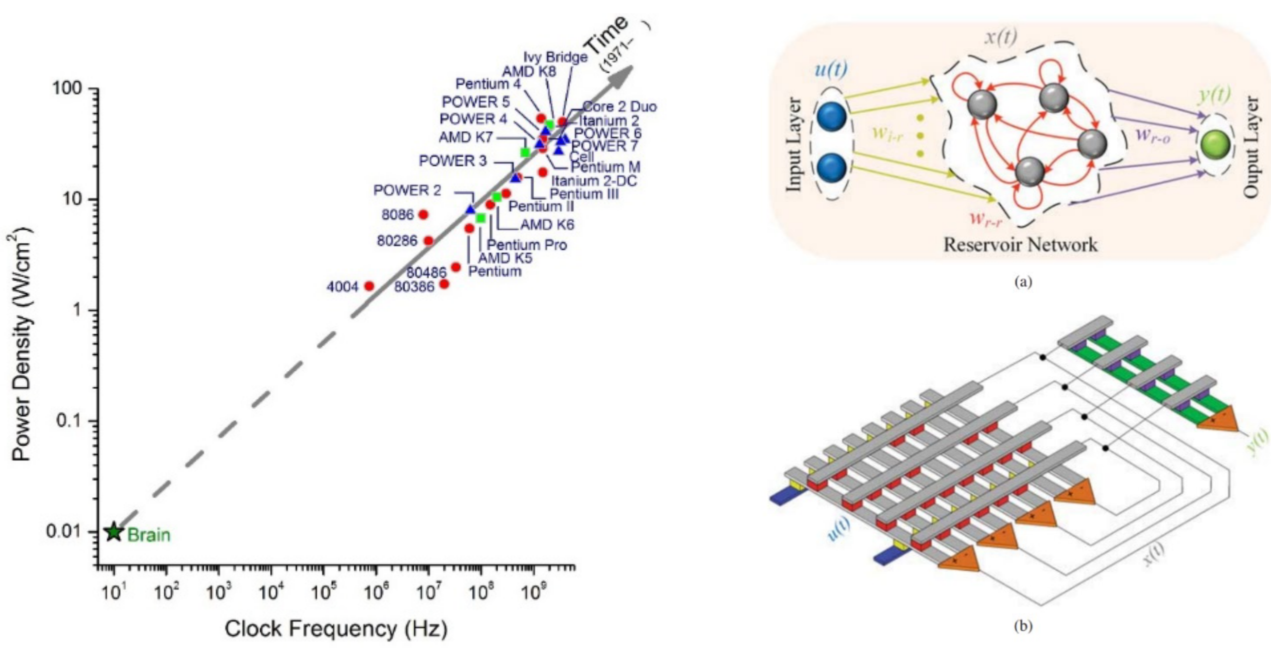
Three educational modules were developed under this work, which largely discussed five different topics related to green computing and sustainable software development:

(a) Information and Communication Technology (ICT) for Green
(b) Green Electronics and ICT
(c) Green Computing and Managing e-Waste
(d) Software Carbon-Footprint
(e) Efficient Design of Datacenters

The first topic is about how we can use current technology in making the environment around us greener, such as smart homes and smart electric grids. The second topic is related to how we can make the technology itself greener, by consuming less energy and exploiting new emerging technologies. The third and fourth topics discuss managing electronic waste from different entities and being able to determine the carbon footprint of a computer algorithm, to eventually minimize the total carbon emissions toll on the environment. Finally, the last topic is related big datacenters and the different techniques used to make them efficient.

One of the educational modules introduced students to emerging power-efficient computing hardware technologies. In this module, students learned about neuromorphic computing systems, which are inspired by the structure and operation of the human brain and are designed to perform computations with significantly lower energy consumption than conventional CPUs and GPUs [18, 19] Fig. 1(a). These systems, illustrated in Fig. 1(b), achieve substantial energy savings for equivalent computational tasks and are particularly well suited for machine learning workloads. The module also covered the use of memristor-based crossbar arrays as a foundational hardware architecture for neuromorphic systems [18]. These architectures enable a highly efficient implementation of large-scale machine learning models and offer a compelling alternative to traditional general-purpose processors. An example of memristor crossbar arrays and their mapping to neural network structures is shown in Fig. 1(b).

Another module focused on green computing and sustainable software development, with an emphasis on quantifying and reducing the carbon footprint of software programs. Students were introduced to methods for estimating software-related carbon emissions and to optimization strategies tailored to specific hardware platforms and programming tasks. Central to this module was the Software Carbon Intensity (SCI) metric developed by the Green Software Foundation [20], as shown in Fig 2,



(a) (b)

Figure 1: Neuromorphic Computing Systems. (a) Power consumption vs. clock frequency of recent devices and the human brain [21]. (b) New emerging crossbar arrays and their implementation of an echo state recurrent neural network [19].

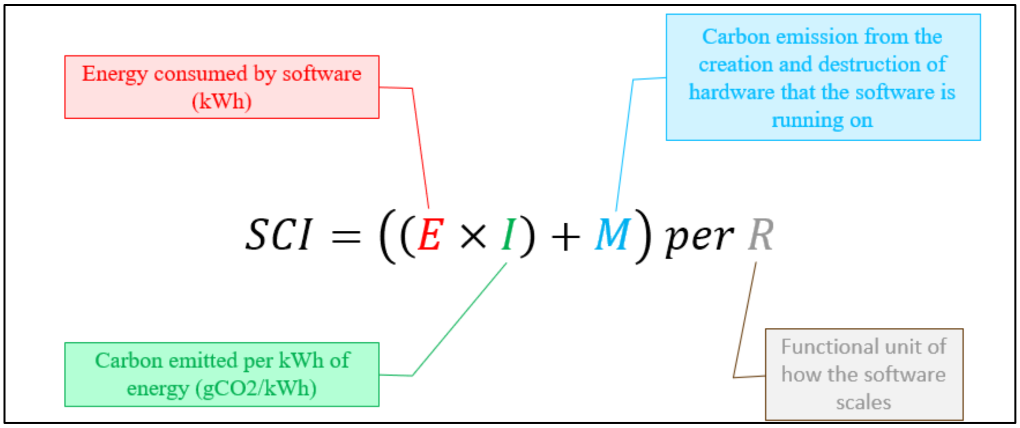


Figure 2: Software Carbon Intensity (SCI) formula for a program.

where E represents the energy consumed by the hardware running the software, in kilowatt-hours, I denotes the carbon intensity of electricity generation in gCO_2/kWh, M corresponds to the embodied carbon emissions associated with the hardware lifecycle, and R is the functional unit describing how the software scales [20]. In addition, students examined the architecture and operation of large-scale data centers, common sources of energy inefficiency, and emerging techniques aimed at reducing energy waste in such environments.

The third module addressed foundational concepts of sustainability and their relevance to computer engineering. Students were introduced to the definition of sustainability and its broader implications for modern technological development, with particular attention to the environmental impact of the Information and Communication Technology (ICT) sector. This module aimed to raise awareness of critical global challenges, such as climate change, and the role that computing technologies play in contributing to these issues. Students were presented with quantitative data on the global carbon footprint of the ICT sector, the typical lifecycle of computing hardware, and ongoing efforts to mitigate environmental impact. An example of the data and visual materials used in this module is shown in Fig. 3, which depicts the amount of energy consumed to produce a single PC versus the per year consumption on a typical workload.

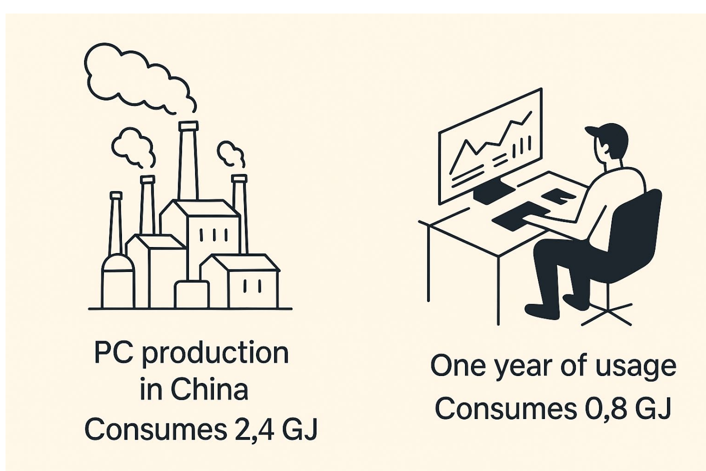


Figure 3: Production versus consumption energy of a single typical PC.

4. Assessment Methods and Results

The educational modules were piloted during Summer and Fall of 2025, targeting students at sophomore, junior, and senior levels. The piloted modules impacted approximately 75 students during these two semesters: 53 in SD and 22 in ABD. We adopt a mix of evaluation methods to assess both learning outcomes and attitudinal shifts. In SD, quantitative evaluation was derived from targeted assignments and performance metrics, directly related to the educational modules proposed, while qualitative data were gathered from ABD through reflective surveys to gauge the students' eco-consciousness and sustainability-oriented technical reasoning.

For the ABD course, most of the students were advanced junior/senior students close to graduation. The newly developed educational modules were offered to the students towards the end of the semester, followed by a post-assessment survey. The survey mainly captures how much the students knew about the sustainability topics discussed before they learned about them in class, and how much the modules changed their view of sustainability. Fig. 4 shows the results for the following question: "*How familiar you were about the different topics discussed in the Green Computing and Sustainable Software Development modules before hearing about them in this course?*" The results show that, at best, only **36%** of the students were at least moderately familiar with the "Green Computing and Managing e-waste" topic. That percentage drops drastically for the other topics, which indicated the gap in knowledge about these topics and how our students lack the sustainability education needed.

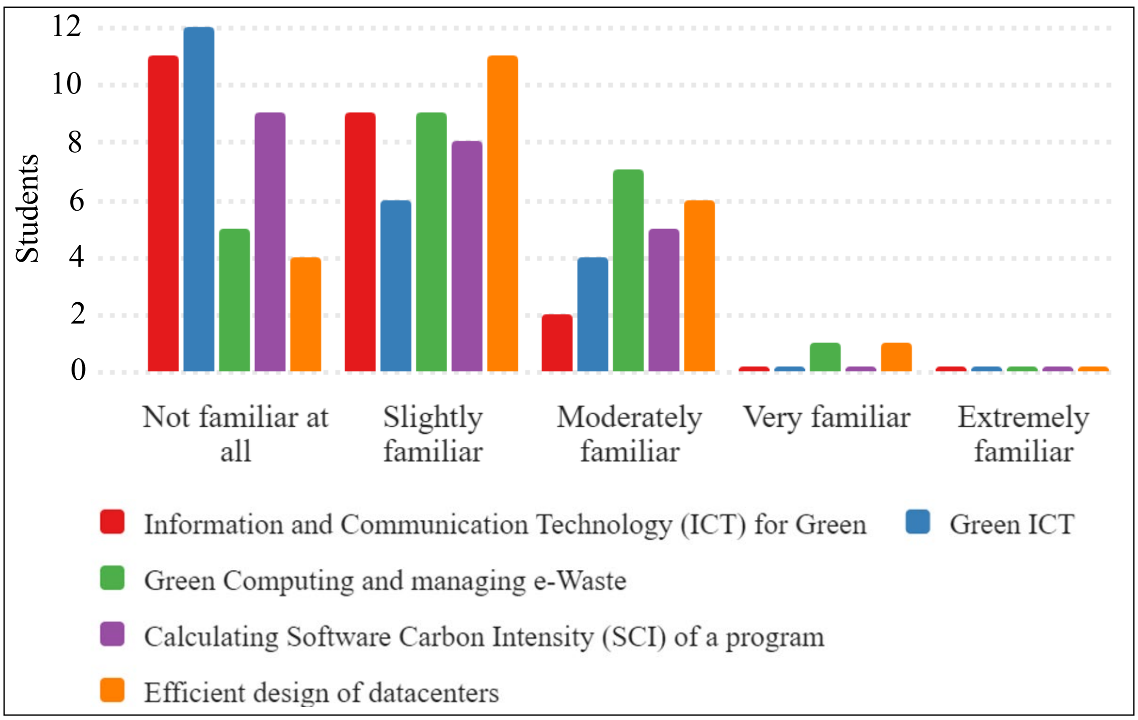


Figure 4: Students' familiarity with the topics before being introduced in the developed educational modules.

The results for the following question: "*After going through this module, how interested you are in knowing more about the topics discussed?*" are depicted in Fig. 5, where **at least 86%** of the students were moderately interested in learning about the topics discussed. In fact, **more than 45%** of the students were "very interested" in learning about topics like "Green Computing and Managing e-waste" and “Calculating Software Carbon Intensity of a Program”. This result does make sense as it is directly related to the course topics on how to handle Big Data.

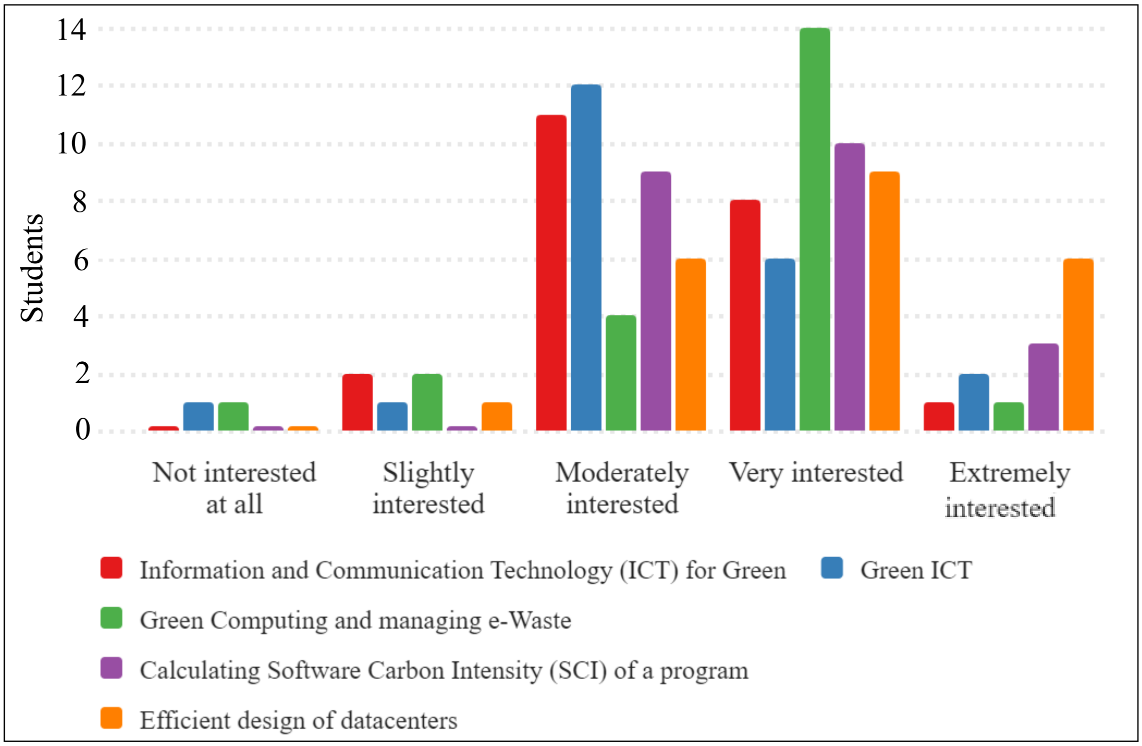


Figure 5: Students’ interest in the topics introduced in the developed educational modules.

Finally, Fig. 6 shows the results for the following question: “*From your own point of view, please rank the following topics by importance:*”, with 1 being the most important and 5 being the least important. The results show that most of the students agree on “Efficient Design of Datacenters” being the most important topic, followed by “Green Computing and Managing e-waste”, which highlights again that they have developed a sense of eco-consciousness towards the sustainability problem and how it is related to Big Data and Generative AI.

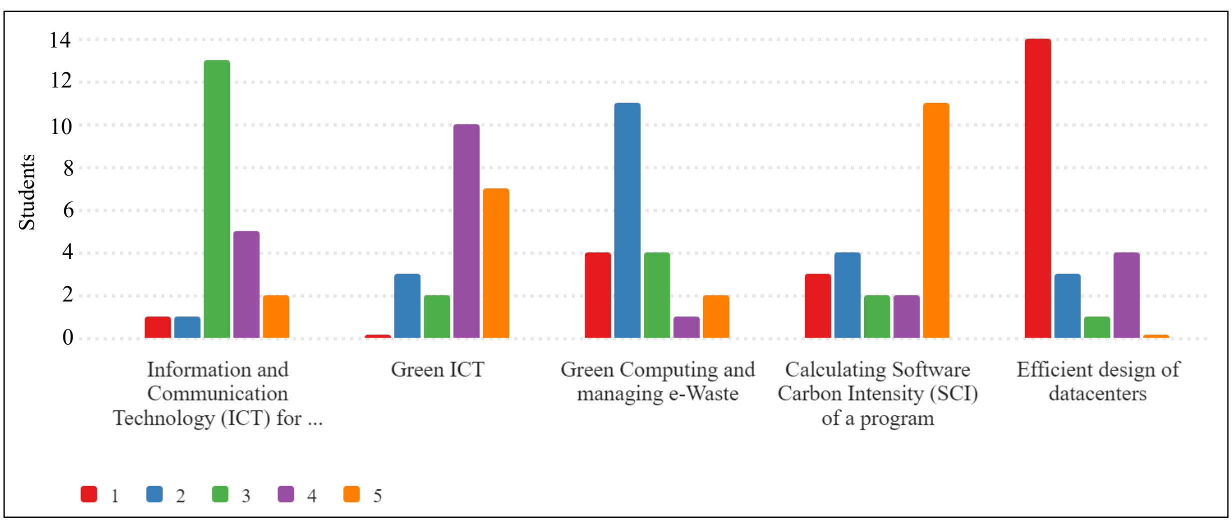


Figure 6: Students’ ranking of the various introduced topics’ importance.

For the SD course, quantitative data were collected from pre and post assignments. Midterm and final reports were used to assess the sustainability aspect of the capstone projects the students submitted. For the latter, the instructor laid out certain sustainability requirements on all the capstone projects that the students designed, while being open-ended enough for the students to innovate. Fig. 7 depicts the analysis of the pre-assessment (a) and post-assessment (b) for 12 teams that were enrolled in the SD course during Fall 2025. Each team had between 4-5 students. The pre-assessment was for the students' proposals for their SD project, which was toward the third part of the semester, and the result of this assessment is in Fig. 7(a). As can be seen, all the teams have considered most the sustainability requirements in their proposal, with about **42%** of them fully implementing all these considerations. Fig. 7(b) shows the results of the post-assessment, which was held at the end of the semester when students fully designed and finished their projects. The percentage of the teams that at least met some of the sustainability considerations in their final design dropped down to about **58%**, with about 42% not meeting any of the requirements at all. The authors are still further analyzing the results and the feedback given by the students to further investigate the reasons that hindered them from implementing the sustainability requirements. But an initial assessment shows that the crunch in time at the end of the semester and the low stakes in the sustainability component of the rubric may be the main reason why some teams decided not to implement it.

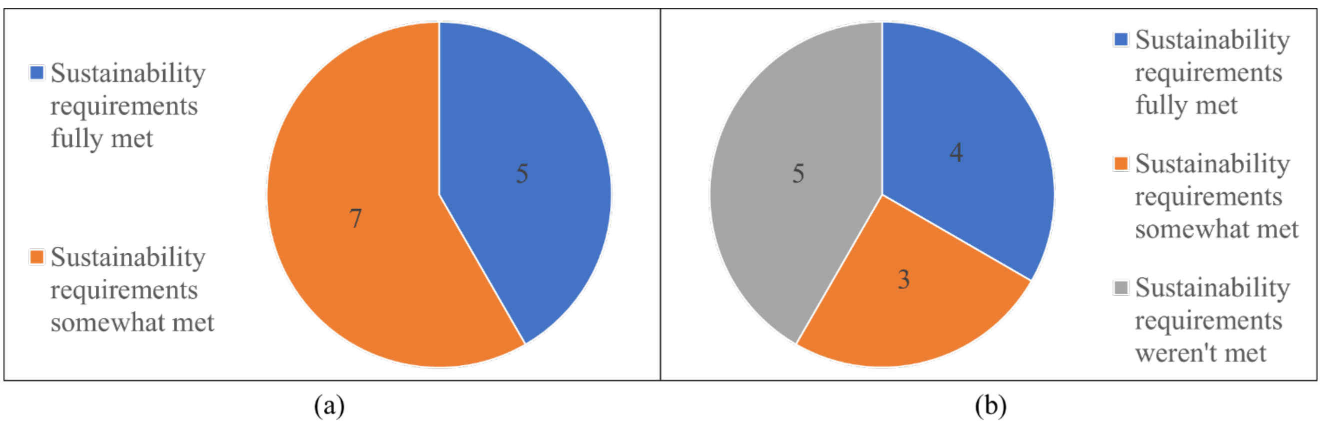


Figure 7: Pre- and post-assessment of the students' engagement with different sustainability considerations in their final capstone project in the Senior Design Course.

5. Summary and Conclusions

In this WIP study, the authors studied interweaving sustainability education in ECE curricula, by offering educational modules discussing different sustainability concepts in two undergraduate courses. Two different evaluation methods were used, and initial results show that students fully engaged in the sustainability educational modules and developed a sense of eco-consciousness towards their profession as engineers who design most of the world's latest technology. Based on the successful implementation of the modules into existing courses, the authors conclude that the most effective response to RQ1 is the development of concise, sustainability-focused educational modules that can be embedded within existing courses. Such modules can be tailored to align with the technical content and learning objectives of the host course, making them more practical and flexible than the creation of stand-alone courses dedicated exclusively to green computing. This modular approach is therefore more efficient for both instructors and students.

As for RQ2, the authors believe that an evaluation plan that mixes both qualitative and quantitative assessments, with more emphasis on the former than the latter, would be the most efficient method to truly capture the impact of the educational modules. A big part of sustainability education is to simply raise the students' awareness of the problem and encourage them to always consider it when they are designing new systems. This is better captured via qualitative assessments (such as pre- and post-surveys, focus groups, interviews, …). While the preliminary results from this study provided insightful information, incorporating additional qualitative assessments such as focus groups and/or interviews into future work would provide even more insights about students' changes in awareness and about gains in knowledge.

Finally, for RQ3, results depicted in Fig. 4 and 5 show that we are clearly lacking in sustainability education, especially in fields related to new emerging technologies, such as LLMs and Generative AI. The authors are still investigating the unexpected results from the post-assessment that was conducted in the SD course, which will be included in future publications related to this work. As indicated above, the use of additional qualitative assessments in future work can provide opportunities to gain a deeper understanding of the results from the other assessments.

For future work, the authors plan to continue offering these educational modules in the same courses but also expand the study and recruit more instructors who are willing to offer them in related courses. After the modules are finalized, the authors are going to publicly disseminate the educational modules, so that instructors, nationally and internationally, can plug-and-play them in their courses as needed.

6. Acknowledgments

We would like to heartedly thank the Lemelson Foundation and ASEE for their financial support of this pilot study.

References

[1] S. Luccioni, Y. Jernite, and E. Strubell, "Power hungry processing: Watts driving the cost of AI deployment?," in *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, 2024, pp. 85-99.

[2] "Why AI uses so much energy — and what we can do about it." https://iee.psu.edu/news/blog/why-ai-uses-so-much-energy-and-what-we-can-do-about-it (accessed 1/2026, 2026).

[3] "AI and ESG: Understanding the Environmental Impact of AI and LLMs." https://www.holisticai.com/blog/environmental-impact-ai-llms (accessed 1/2026, 2026).

[4] "Green500." https://top500.org/lists/green500/ (accessed May 2024.

[5] A.-K. Peters *et al.*, "Sustainability in computing education: A systematic literature review," *ACM Transactions on Computing Education,* 2023.

[6] D. C. Chou, H.-G. Chen, and B. Lin, "Green IT and corporate social responsibility for sustainability," *Journal of Computer Information Systems,* vol. 63, no. 2, pp. 322-333, 2023.

[7] S. Abdulsalam, Z. Zong, Q. Gu, and M. Qiu, "Using the Greenup, Powerup, and Speedup metrics to evaluate software energy efficiency," in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, 2015: IEEE, pp. 1-8.

[8] B. W. Ford and Z. Zong, "Portauthority: Integrating energy efficiency analysis into cross-platform development cycles via dynamic program analysis," *Sustainable Computing: Informatics and Systems,* vol. 30, p. 100530, 2021.

[9] M. Couto, J. Saraiva, and J. P. Fernandes, "Energy refactorings for android in the large and in the wild," in *2020 IEEE 27th international conference on software analysis, evolution and reengineering (SANER)*, 2020: IEEE, pp. 217-228.

[10] R. Morales, R. Saborido, F. Khomh, F. Chicano, and G. Antoniol, "Earmo: An energy-aware refactoring approach for mobile apps," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 59-59.

[11] S. Hasan, Z. King, M. Hafiz, M. Sayagh, B. Adams, and A. Hindle, "Energy profiles of java collections classes," in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 225-236.

[12] S. Imperatives, "Report of the World Commission on Environment and Development: Our common future," *Accessed Feb,* vol. 10, pp. 1-300, 1987.

[13] J. Saraiva, Z. Zong, and R. Pereira, "Bringing green software to computer science curriculum: perspectives from researchers and educators," in *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 2021, pp. 498-504.

[14] D. Torre, G. Procaccianti, D. Fucci, S. Lutovac, and G. Scanniello, "On the presence of green and sustainable software engineering in higher education curricula," in *2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM)*, 2017: IEEE, pp. 54-60.

[15] C. Pang, A. Hindle, B. Adams, and A. E. Hassan, "What do programmers know about software energy consumption?," *IEEE Software,* vol. 33, no. 3, pp. 83-89, 2015.

[16] I. Manotas *et al.*, "An empirical study of practitioners' perspectives on green software engineering," in *Proceedings of the 38th international conference on software engineering*, 2016, pp. 237-248.

[17] G. Pinto, F. Castor, and Y. D. Liu, "Mining questions about software energy consumption," in *Proceedings of the 11th working conference on mining software repositories*, 2014, pp. 22-31.

[18] A. M. Hassan, C. Yang, C. Liu, H. H. Li, and Y. Chen, "Hybrid spiking-based multi-layered self-learning neuromorphic system based on memristor crossbar arrays," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 2017: IEEE, pp. 776-781.

[19] A. M. Hassan, H. H. Li, and Y. Chen, "Hardware implementation of echo state networks using memristor double crossbar arrays," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017: IEEE, pp. 2171-2177.

[20] "The Green Software Foundation." https://greensoftware.foundation/ (accessed May 2024).

[21] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science,* vol. 345, no. 6197, pp. 668-673, 2014.