WIP: Introducing Green Computing and Sustainable Software Development in Computer Engineering Curricula

Amr M. Hassan
ECE Department
University of Pittsburgh
Pittsburgh, PA, USA
amm418@pitt.edu

Mohamed A. S. Zaghloul
ECE Department
University of Pittsburgh
Pittsburgh, PA, USA
mab433@pitt.edu

***Abstract*—In this Research-To-Practice WIP paper, the authors are going to introduce educational modules about green computing and sustainable software development in an elective course, named Algorithms for Big Data, which is taught at the Electrical and Computer Engineering Department hosting this study. The initiative is aiming to educate undergraduate computer engineers about the negative impact of the software industry on the environment and how to design sustainable solutions for different sectors of this industry. The impact of the new modules on students is assessed via a survey and the results show that students find the modules to be interesting and enriched their knowledge about the problem. The authors will also share the best methods of incorporating green computing and sustainable software development in undergraduate engineering curricula.**

Keywords—computer engineering, green computing, sustainable software development.

I. Introduction

It is now clear that climate change is a reality, and the energy crisis is becoming more severe every day. As a result, societies must prioritize eco-innovation and sustainable practices like never before. We, as educators, carry the huge responsibility of enlightening the future generation of engineers about this crisis, and how to identify, approach, and solve sustainability problems in every engineering field. One of the major problems that has a significant negative impact on the environment is the current practices in computing and software engineering. Some examples of this include electronic waste generated by outdated computers due to software upgrades, as well as increase in the energy requirements [1] of newer software versions, especially with the booming of generative Artificial Intelligence (AI) and the huge number of computing resources it consumes [1].

Green and sustainable software engineering is a rising field which aims at adopting energy-efficient software and hardware in an environmentally friendly manner with less negative impacts and carbon footprint. In fact, this topic is very important now that there is a list of the top 500 green super computers in the world, updated semi-annually, to promote the use of them [2]. Research in this field is progressing rapidly during the past decade to come up with green data structures and libraries [3], sustainable software development practices [4, 5], developing measuring metrics [6, 7], and Green IT [8, 9]. In a broader sense, green computing refers to the methods of creating, utilizing, and disposing of computing resources in an eco-friendly manner that preserves overall computing functionality [10]. This encompasses minimizing the use of harmful substances, optimizing product performance throughout its lifespan while minimizing its energy usage, and ensuring that the used resources are either reusable, recyclable, or biodegradable to reduce their environmental impact [10].

Computer engineering programs have been slow in incorporating this education into their curricula, as can be told by the scarcity of the academic publications in this field [11]. This leaves a lot of Research Questions (RQs) to be answered, such as: 1) how can we interweave educational modules about this topic in a technical course, 2) what contexts/applications should these modules address, and 3) how can we assess the impact of these modules on students learning and eco-consciousness about the problem?

In this work-in-progress, we address these research questions via incorporating educational modules that teach students about green computing and sustainable software development in a computer engineering elective course, offered by the Electrical and Computer Engineering (ECE) Department at the University of Pittsburgh. The initial round of assessment will be using a survey to capture the students perspectives and opinions on the impact of the new modules in changing their perspective about green computing and adopting it.

The rest of the paper is organized as follows: Section II briefly mentions previous work in this area. Section III introduces the course under investigation and the details of the newly created modules. Section IV provides the assessment methos used and a discussion on the obtained results, while section V concludes the paper.

II. Background and Previous Works

Although the research in green computing and sustainable software development is advancing rapidly, recent articles show that software engineers still lack the technical background and sustainability education to design greener software [12, 13]. In 2016, Pang *et al.* [12] surveyed Reddit programmers about their knowledge of green software development. The findings indicated that programmers had little-to-no understanding of energy efficiency, lacked familiarity with optimal strategies to minimize software energy consumption, and were uncertain about software energy consumption, which indicated the urgent need for

education and training in this field [12]. Similar concerns were also raised by Pinto *et al.* in [14], when they noticed that not only the number of software energy consumption related questions posted on Stack Overflow had risen quickly in the past decade, but also the answers to them reflected very poor knowledge of the field.

On the other hand, other several studies shared similar concerns about the lack of this type of education in academia. In 2021, Saraiva et al. conducted a research study with 21 well-known researchers and educators in the field of green computing and sustainable software development [11]. The study concluded the lack of courses and educational materials for teaching green computing in higher education, especially computer science curricula [11]. They also highlighted the major challenges behind this result and concluded that, while developing independent courses in green computing or sustainable software development is ideal, it is a very hard task considering the lack of resources and expertise [11]. It's better to incorporate mini-educational modules scattered through different related courses to disseminate this knowledge [11].

In addition, Damiano *et al.* also surveyed 33 academic professors about existing educational modules related to green computing and sustainable software development in higher education [15]. The results obtained showed that this kind of education is lacking behind due to many reasons, including lack of teaching materials; lack of the awareness needed; and lack of technology and the tools needed to support it [15].

Last but not least, in 2023, Peters *et al.* analyzed 89 different, but relevant, publications to sustainability in computing education [8]. The paper summarized the scattered efforts in incorporating green computing and sustainability related education from four different points of view: pedagogical methods used, topics discussed, building blocks, and learning outcomes. The results indicated that after the students were introduced to green computing education, they had developed higher levels of environmental awareness of the impacts of the Information and Communication Technology (ICT) industry in general, mindset change, and Increased motivation towards approaching the sustainability problem for the greater good [8]. The authors also concluded with the need to combine our efforts as engineering educators towards more infrastructural change in computer engineering curricula to adopt these new topics.

Consequently, the authors of this work felt very motivated to start the sustainability education initiative discussed in the following section, in hopes that we can spark similar initiative across other academic institutions, both domestically and internationally.

III. Course Format

The course in which this study is conducted is named Algorithms for Big Data and it has two distinct segments. First, students receive theoretical education on different algorithms than can analyze and handle big data. Afterwards, they apply what they have learned by implementing four labs that are based on Microsoft Azure® cloud computing services, where they can build different processing pipelines, optimize them, and analyze big datasets. Because of the nature of this course in requiring huge cloud resources and computational power to handle that type, it makes perfect sense to introduce the students to the concept of green computing and sustainable software development to make their implementation energy efficient and eco-friendly.

The students also have seven homework assignments throughout the whole semester that include both analytical and programming problems and 4 lab reports that they need to prepare after completing each lab. These assignments are worth 40% of their final grade. They also have a semester long project where they must choose between one of two categories. First, a programming project where they explore a big data algorithm that wasn't discussed in class, implement it, and analyze any publicly available big dataset using it. The other category is a survey/research project where they explore state-of-the-art advances in any topic that is related to big data by reading at least five high quality research papers. Then, students from both categories write a final report in IEEE journal proceeding format to summarize their findings and do a 15-minute presentation in front of the whole class. The project (presentation) is worth 40% (10%) of their final grade. The last 10% of their final grade are given according to their in-class participation throughout the semester.

A. Theoretical and Labs Segements

The course presented in this work is a semester-long undergraduate course. At the university where this study took place, the course is an elective for both computer engineering and electrical engineering majors. The course is typically taken by students in their junior or senior year, after they have completed two core courses in data structures and probability.

In this course, students learn about big data, its various applications, and different mathematical models and algorithms to efficiently analyze it. Some of the topics that are discussed in this segment are: randomized streaming algorithms (such as Morris algorithm, frequency of moments theory and its applications, Bloom filters, count and countmin sketches, etc…); fast matrix multiplication for exact and approximate solutions; finding similar sets (including shingling, min-hashing, and locality-sensitive hashing); and MapReduce platform and its different implementations.

For the lab segment, students use Microsoft Azure® to build cloud-based solutions for different problems. one of the projects they work on is using Azure Machine Learning (ML) Studio to train many predictive ML models, compare between them, and build a predictive web service that can published and accessed from anywhere. The learning objective behind this lab is to know how to optimize the pipeline to use the least cloud resources and computing nodes. Another lab is to utilize Azure Natural Language Processing platforms to optimize a pre-built Large Language Model (LLM) for specific tasks.

B. Green Computing and Sustainable Software Development Educational Modules

The authors developed three different educational modules targeting sustainability in computer engineering to be introduced to the students at the last two classes of the semester. The modules discussed three important topics,

which are 1) sustainability definition and how it is related to the computer engineering field, 2) green computing and sustainable software development, and 3) new emerging power-efficient computing hardware systems. For topic 1, students learn about the definition of sustainability in general and why it is important to think about our planet and how modern industries are affecting it. This is important to raise students' awareness about major sustainability problems, like global warming for example, and how the Information and Communication Technology (ICT) sector is playing a big role in it. Students are presented first with different facts about ICT sector carbon footprint around the globe, the typical life cycle of a regular ICT computing node, and what are the current efforts to neutralize this bad effect. Figure 1 shows an example of the facts that are shared with the students for this module.

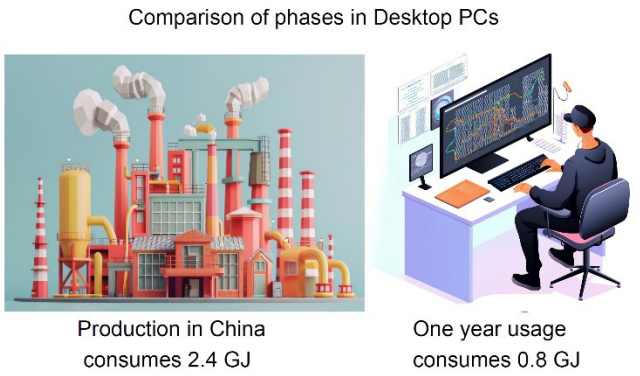

Fig. 1. The energy needed to make a typical PC versus the energy consumed during one year of usage.

In topic 2, students are introduced to the concept of the carbon footprint of a software program, how to estimate it, and how to optimize it (given a specific hardware and specific coding problem). Students learn about the Software Carbon Intensity (SCI) equation, which is developed by the Green Software Foundation [16]. The SCI emission rate equation is displayed by (1):

$$SCI=((E\times I)+M) \ \ per\ R \tag{1}$$

Where E is the energy consumed by software in kWh, I is the carbon emitted per kWh of energy, in gCO_2/kWh, M is the embodied carbon emission from the creation (and destruction) of hardware that the software is running on, and R is the functional unit of how the software scales [16]. In the same module, students also learn about big datacenters, their architectures and layouts, how they waste energy, and the novel techniques that are used to minimize energy wasted in them.

Finally in topic 3 module, students learn about new emerging computing devices that are designed to mimic the human brain, called neuromorphic computing systems [17-19], which requires folds of energy less than what a typical CPU or GPU would consume for the same task (depicted in Fig. 2). These computing systems are based on Memristor crossbar arrays [18] and are optimized to implement large ML models very efficiently, compared to general purpose CPUs and GPUs. Figure 3 shows an example of Memristor crossbar arrays and how they are used to mimic a typical neural network.

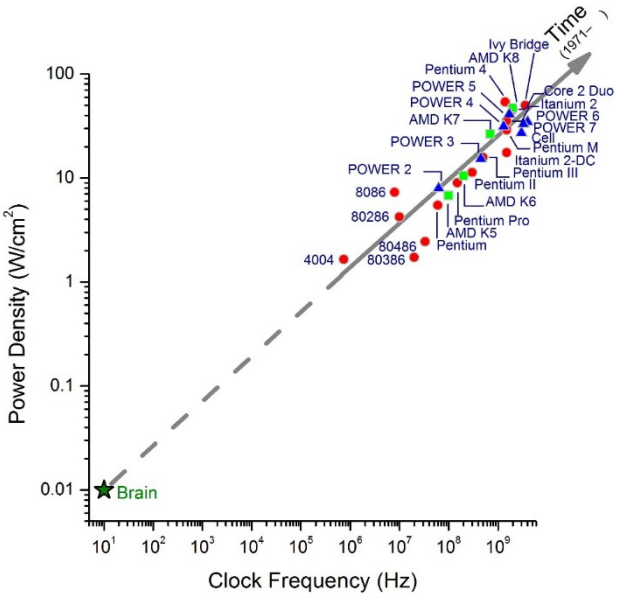

Fig. 2. Power consumption vs. clock frequency of recent devices and the human brain [20].

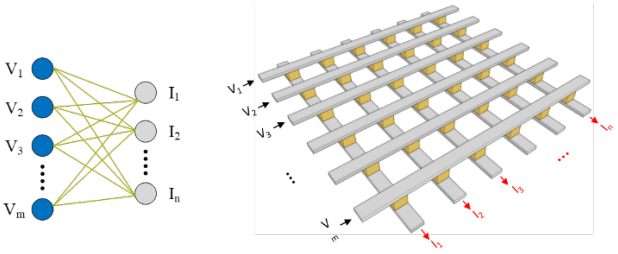

Fig. 3. Memrisyor crossbar arrays and hwo they are used to mimic a neural network model [18].

IV. Assessment Methods and Results

For assessment, students were surveyed over the recent offering of this course during Fall 2023 to capture their perspectives and opinions on the impact of the new modules on their learning. The survey was conducted at the end of the semester and a Complete/Incomplete mark was given to the students accordingly. IRB approval was obtained before carrying out the study. The class had seven students and all of them completed the survey (n = 7). As shown in Table 1, the survey had five questions, where two of them were rated on a five-point Likert-type scale (Q1 was to capture their familiarity with the topics before taking the new educational modules and Q2 was to capture their interest in them). Q3 was multiple choices to see which topics they were interested about, while Q4 was to see if they like to dedicate more class time to know about green computing. The last question was open-ended to know their opinion about any possible improvements to the modules. The results for all the questions, except Q5, are shown in Table 1 under each corresponding answer. As can be told from the results of Q1 and Q2, students were not familiar about these important

TABLE 1. ASSESSMENT OF THE NEW LEARNING MODULES FOR GREEN COMPUTING

Q1 - 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?	*Not familiar at all* **29% (2)**	*Slightly familiar* **71% (5)**	*Moderately familiar* **0%**	*Very familiar* **0%**	*Extremely familiar* **0%**
Q2 - After taking these modules, how interested you are in knowing more about the topics discussed in that modules?	*Not interest at all* **0%**	*Slight interest* **0%**	*Moderate interest* **71% (5)**	*High interest* **29% (2)**	*Extreme interest* **0%**
Q3 - Which of the topic(s) that were discussed would you like to know more about?	*ICT for Green* **14% (1)**	*Green ICT* **14% (1)**	*Green Computing & managing e-waste* **43% (3)**	*Carbon footprint of algorithms* **100% (7)**	*Efficient design of datacenters* **57% (4)**
Q4 - Would you prefer to have more class time to discuss these topics in more details or even discuss different topics related to the same problem?		*Yes* **43% (3)**	*Maybe* **57% (4)**	*No* **0%**	
Q5 - What are your suggestions on improving this module for future offerings (on any possible aspect of it)?					

topics and their impact on the environment, and when they did learn about them, they developed great interest in knowing even more about the problem and how to solve it.

For Q3, 100% of the students were interested in knowing more about the carbon footprint of algorithms and how to calculate and minimize it. Then, about half of the students were interested in minimizing the carbon footprint from hardware that is related to the ICT industry in general. Q4 result shows that students would like to have more class time dedicated to green computing and sustainable software development. These results show the great interest that computer engineers easily developed towards the sustainability problem when they are introduced to it with context related to their field of expertise.

Finally, the answers of Q5 can be summarized in the students wanting to have more practical implementations of these sustainability related topics and how it can be related/applied to their future jobs. They also wanted to know more about industry standards of sustainability topics in the ICT field. In addition, they wanted to see different examples of calculating the carbon footprint on different algorithms and analyze the impact of optimizing them on the environment.

V. Conclusion and Future Work

In this work-in-progress, the authors shared their experiences in interweaving educational modules about green computing and sustainable software development in an elective class targeting computer engineering majors. They also shed light on how important it is to start incorporating such education into computer engineering and computer science curricula. Based on the results obtained in this survey and from the literature review, the authors believe that the best answer to RQ1 is to design small educational modules related to green computing that can be integrated into existing courses, where these modules can be adapted to the courses they are taught in. This would be more efficient for both the instructors and students than creating full courses targeting this topic. Moreover, judging by the results obtained from answering Q3 and Q5 above, the authors thinks that the answer to RQ2 would be to discuss sustainability issues directly related to the ICT industry, as well as any other industry that can be partially related to it.

On the other hand, RQ3 is only partially answered in this WIP, judging by the interest the students showed in knowing more about these topics and the increased awareness about the problem, which is a direct result of teaching the new educational modules. However, the authors still need to do quantitative assessment to measure how deeply the students understand the problem and can apply different innovative techniques to minimize the carbon footprint of their designed systems.

As future work, the authors will be building up on this work by modifying the educational modules to: 1) accommodate students request form the survey to incorporate more technical examples, 2) dedicate more in-class time to talk about these topics, and 3) introduce a design problem where students need to come up with an innovative and sustainable solution for it, based on their educational background and the diverse environments they were raised in. The latter is meant to promote diversity, equity, and inclusion (DEI) and highlight how DEI can improve the quality of the produced sustainable solutions. These new modules will be introduced in the same course and will be rolled out in Summer and Fall 2024 and a follow-up paper will also be published on the findings from this study.

Acknowledgment

The authors would like to acknowledge the financial support from the Mascaro Center for Sustainable Innovation (MCSI) at the University of Pittsburgh this study and thank them for their continuous support and endeavor towards a greener world for all mankind.

References

[1] G. Pinto and F. Castor, "Energy efficiency: a new concern for application software developers," *Communications of the ACM,* vol. 60, no. 12, pp. 68-75, 2017.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[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] B. Yan, A. M. Mahmoud, J. J. Yang, Q. Wu, Y. Chen, and H. H. Li, "A neuromorphic ASIC design using one-selector-one-memristor crossbar," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016: IEEE, pp. 1390-1393.

[20] 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.