Innovations in Systems and Software Engineering
https://doi.org/10.1007/s11334-023-00548-9

SI: CARBON AWARE COMPUTING SYSTEMS

A holistic approach to environmentally sustainable computing

Andrea Pazienza[1] · **Giovanni Baselli**[1] · **Daniele Carlo Vinci**[1] · **Maria Vittoria Trussoni**[1]

Received: 1 August 2023 / Accepted: 9 December 2023
© The Author(s) 2024

Abstract

Placing sustainability at the core of computing practices, the industry is poised to pioneer positive changes and create a cleaner and more sustainable world for future generations. The environmentally sustainable computing (ESC) framework is introduced in this paper as an innovative solution to revolutionize sustainability practices across various computing domains and cover multiple aspects of sustainable information technology (IT). The ESC framework includes the entire lifecycle of computing systems, including critical stages such as design, development, monitoring, refactoring, and regulatory compliance. Through the adoption of the ESC framework, academia and industry stakeholders can gain a powerful tool to evaluate and measure sustainability factors across different computing domains and can integrate eco-friendly computing principles and patterns throughout their products and services. This can significantly reduce their carbon footprint while complying with environmental regulations. In addition to presenting the ESC framework, the paper showcases real-world use cases. The first involves a leading Italian bank, emphasizing the significance of monitoring and compliance in achieving sustainable solutions within carbon-aware computing. The second use case explores resource efficiency optimization in Kubernetes clusters, illustrating how the ESC framework aligns with cloud infrastructure management trends.

Keywords Carbon-aware Computing · Environmentally Sustainable computing · Sustainability · Green IT

1 Introduction

The quest for sustainability has emerged as an imperative goal in the ever-evolving landscape of modern business and computing technologies. The increasing awareness of environmental challenges, resource constraints, and the impact of human activities on the planet has led organizations to rethink their strategies and operations. Carbon-aware computing has garnered significant attention in this context as a pivotal avenue for advancing environmental stewardship within the digital age. The rationale for pursuing carbon-aware computing is deeply rooted in the recognition of the IT industry's profound influence on the global carbon footprint and resource consumption. As IT continues to permeate every facet of our lives, the exponential growth of computational power in data centers, cloud computing, electronic devices, and digital services has amplified the energy demands and greenhouse gas (GHG) emissions associated with IT operations. The significance of this impact has necessitated a shift toward responsible and eco-conscious practices, forging a path toward a greener and more sustainable future.

According to [1], it is estimated that the IT sector's carbon footprint is 730 Mt CO_2-equivalents (CO_2eq) or 1.4% of overall global emissions, pointing that these values are bound to grow more and more. In a more recent work [2], the true proportion of the IT sector's share of global greenhouse emissions could be around 2.1–3.9% instead of previous calculations that did not account for the full life-cycle and supply chain of IT products and infrastructure.

Therefore, the urgent need to address the environmental impact of computing technologies has led to the emergence of the Environmentally Sustainable Computing (ESC) paradigm, also known as Sustainable IT or Green IT, which seeks to develop eco-friendly solutions. The theme of ESC is of utmost importance in the context of the current global climate crisis. The exponential growth of computing infras-

✉ Andrea Pazienza
andrea.pazienza@nttdata.com

Giovanni Baselli
giovanni.baselli@nttdata.com

Daniele Carlo Vinci
danielecarlo.vinci@nttdata.com

Maria Vittoria Trussoni
mariavittoria.trussoni@nttdata.com

[1] Innovation and Advanced Technology, NTT DATA Italia S.p.A., Via E. Calindri, 4, 20143 Milan, Italy

Published online: 06 February 2024

tructures and their contribution to GHG emissions necessitate a concerted effort to promote energy-efficient practices across the computing landscape. While existing research has made significant strides in addressing isolated aspects of sustainability, the complexity and interconnectedness of computing systems call for a holistic approach. Thus, our aim is to fill the gap in the literature by proposing a unified framework that encompasses diverse aspects of IT sustainability and facilitates a comprehensive understanding of the challenges and opportunities in ESC.

Creating an inclusive framework is compulsory for several reasons. First of all, the field of IT is characterized by a vast array of technologies, applications, and systems, each with unique environmental implications. Sustainable data centers, eco-friendly coding practices, sustainable web design, sustainable artificial intelligence (AI), and many others are domains that contribute to the overall environmental impact of computing technologies. A holistic framework would allow us to comprehensively address and integrate these various aspects, capturing the full breadth of IT solutions related to sustainability. Also, it involves experts from multiple fields, including computer science, data center management, AI, software development, and environmental sciences. An extensive and shared framework would encourage collaboration and knowledge sharing among specialized experts, fostering a cohesive effort toward environmentally sustainable practices across different computing disciplines, and facilitating a more integrated approach, fostering researchers and professionals to consider the interconnectedness and interdependencies between various sustainability dimensions. Moreover, different computing domains have distinct environmental footprints and challenges. A holistic framework would ensure that each domain's unique sustainability goals are adequately addressed, promoting a balanced approach toward mitigating the environmental impact of IT.

In this paper, we propose a comprehensive framework for ESC that spans the entire lifecycle of computing systems. Our framework comprises three key dimensions: new releases, running systems and governance. These three options are further divided into Design, Development, Monitoring, Refactoring, Accounting and Reporting, Regulations, and the last one of Innovation, Mindset, and Way of Working. We envision that the adoption of the ESC framework will lead to substantial improvements in both the academic and industrial realms. In academia, researchers will benefit from a structured approach to study, quantify, and monitor sustainability factors across computing domains. This will lead to a more rigorous evaluation of existing solutions and the development of novel strategies to enhance sustainability. In the industrial sector, the ESC framework will empower businesses to implement sustainable computing practices at every stage of product development and deployment. By integrating sustainability into the core of their operations, industries can mitigate their carbon footprint, adhere to regulatory requirements, and foster a positive environmental impact.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of the background and related work in carbon-aware computing and sustainable computing practices. In Sect. 3, we detail the methodology underlying the ESC framework, delineating each dimension and its significance. Section 4 presents the real-world use case developed within the banking sector, where a solution of the framework is applied to a specific carbon footprint monitoring scenario, showcasing its practical implications. Section 5 delves into the implications of standardization and validation processes about the ESC framework. Finally, Sect. 6 concludes the paper, outlining future work.

2 Background and related work

Despite the extensive technological development we encounter every day, the life cycle of IT equipment is associated with several negative environmental impacts [3]. The generation of e-waste is expected to increase to 74.7 Mt in 2030 and reach 110 Mt in 2050 unless we change the practices for its management [4]. In addition, the operation of computers, networks, and data centers involves the consumption of large amounts of electricity, followed by the production of huge amounts of CO_2. At the same time, IT and information systems (IS) are crucial factors driving the transformation to a more sustainable economy and society. Both the academic community and practitioners have identified a wide range of areas where IT can reduce negative environmental impacts, particularly through the re-engineering of business and manufacturing products and processes. In addition to improving the efficiency of internal processes, energy monitoring and environmental management systems facilitate transparency and enable measurement of the achievement of environmental goals. These contradictory effects are discussed by the IT research community under the name of environmentally sustainable computing, often referred to as sustainable IT or green IT.

The concept of sustainable IT has been around for several years, but it has gained increasing attention in recent years due to the growing awareness regarding sustainability and the environment. First of all, the term sustainable refers to technologies and processes that are environmentally friendly, namely that have a less negative impact on nature than conventional systems. Specifically, when referring to the environmental impact of sustainable technologies, actually we refer to the environmental footprint associated with their life cycle, that is the quantification of a product's potential environmental impacts through specific impact identifiers such as GHG emissions. On the other hand, we are aware of the environmental impact of sustainable processes when we refer to finding ways to reduce resource inputs, decrease pollution, and reuse materials in all those activities involved in the production of goods or services, including the processes of raw material extraction, industrial processing, distribution and disposal of finished products or waste. Thus, the objective of sustainable IT concerns all measures and initiatives that reduce the negative environmental impact of the production, operation, and disposal of IT equipment and infrastructure, and includes practices that determine the investment, implementation, use, and management of information systems to minimize the negative environmental impacts of information systems, business operations, and IS-based products and services [5].

A further definition is provided by the Green Software Foundation (GSF), which defines Sustainable IT as an emerging discipline at the intersection of climate science, software design, electricity markets, hardware, and data center design. GSF was created by software businesses and practitioners to build a trusted ecosystem of people, standards, tooling, and best practices for creating and building sustainable software. The Foundation's mission is to reduce the total change in global carbon emissions associated with software. The GSF has created a metric called the Software Carbon Intensity Specification (SCI) which defines a methodology for calculating the rate of carbon emissions for a software system. The SCI is also measured in terms of CO_2eq. The purpose is to help users and developers make informed choices about which tools, approaches, architectures, and services they use in the future. Since the SCI is created by software practitioners, it is a better metric for people building and operating software to measure carbon footprint and take actions to reduce it. For this purpose, GSF is concerned with defining a new set of principles, patterns, and practices. Principles are important because they provide fundamental guidance for the development of sustainable software solutions, helping to minimize environmental impact and maximize benefits to the environment and society. In Table 1 have been summarized all the principles to date recognized and defined by GSF.

In particular, the first principle goal is to emit as little carbon (GHG) as possible per unit of work. This is because carbon contributes to global warming, so much so that the international community wants to limit its impact through actions and agreements such as the Paris Climate Agreement, agreed by the United Nations in 2015 with the goal of reaching the 1.5° lower limit of global temperature increase.

Energy efficiency and carbon efficiency are highly correlated, as electricity is a carbon proxy [6]. Developing green software, therefore, means producing software that consumes the least amount of energy. It is important to measure, quantify the energy consumption of an application, and also evaluate the efficiency of hardware use, which should be used as much as possible.

The energy used has different impacts based on carbon intensity. The latter varies depending on where and when the relative energy is used. Consuming energy when the carbon intensity is low allows more renewable energy to be used and, as a result, reduces environmental impact. To reduce one's impact even more, there are techniques for shifting energy demand to places (space shifting) and/or times (time shifting) with lower carbon intensity. Demand shaping, on the other hand, allows energy consumption to be adjusted to the variability of carbon intensity to consume more in low-intensity periods and less in high-intensity periods.

The carbon emitted during the entire life cycle of a device is called embodied carbon. To calculate total pollution, it is necessary to consider both that emitted during the operation of the electronic device and the embedded carbon associated with its creation and disposal. Extending the lifespan of a device has the effect of amortizing the carbon emitted, thus reducing its annual impact. Using cloud computing, for example, is more energy efficient than an on-premise server because of the ability to apply techniques such as demand shifting and demand shaping [7, 8].

Various methodologies exist for quantifying impact, and one prominent tool employed globally is the GHG protocol, designed for organizations to gauge their overall carbon emissions. The GHG protocol classifies carbon emissions into three scopes, encompassing both direct and indirect emission categories. However, the application of the GHG protocol encounters challenges when assessing emissions from open-source software. To address this, the SCI metric was devised as an alternative for quantifying software-generated emis-

Table 1 GSF's principles of sustainable IT

Principle	Description
Carbon efficiency	Emit the least amount of carbon possible
Energy efficiency	Use the least amount of energy possible
Carbon awareness	Do more when electricity is cleaner and do less when it is dirtier
Hardware efficiency	Use the least amount of embedded carbon
Measurement	What you can't measure, you can't improve
Climate commitments	Understand the exact mechanism of carbon emission reduction

sions. Unlike a cumulative measure, SCI is expressed as a score, introducing the concept of functional units (e.g., number of users, ML jobs, etc.). The determination of these functional units is not prescribed by SCI and should be tailored based on the specific characteristics and applications of the software in question.

There are three general categories of methodologies to combat climate change: abatement, offsetting, or neutralization. Abatement is the most effective way to combat climate change and includes increasing energy efficiency. Offsetting refers to the adoption of renewable energy sources, sustainable living practices, recycling, etc., whilst neutralization refers to the removal and permanent storage of atmospheric carbon. An organization can call itself carbon neutral when its total emissions match the total emissions offset through carbon reduction projects. In contrast, the goal of net zero is to abate emissions and offset only those residual emissions that cannot be eliminated. SCI is designed for emissions elimination and a separate neutralization strategy to form the basis of a net-zero strategy for an organization.

Gathering and raising awareness concerning each of these principles makes it possible to produce new technological solutions that can effectively reduce one's impact on the environment or solutions that can support sustainability. In fact, there are two main concepts behind any sustainable IT project:

- Sustainability by IT: the development of sustainability-focused systems that can assist in supporting the achievement of certain requirements related to the environmental sphere (systems to track companies' carbon emissions or to help them reduce water consumption);
- Sustainability of IT: the development of inherently sustainable systems, impacting the environment as little as possible throughout their value chain. This includes evaluating emissions related to the energy mix.

The concept of IT impact on sustainability can also take on different connotations. It can be described through three different aspects:

1. Footprint (Sustainability of IT): it refers to the negative impact of IT services on the environment and the activities required to reduce that. Typical activities include:
    - Reducing the carbon footprint of business applications
    - Maximizing circularity of IT hardware
    - Optimizing Data Centre energy efficiency
    - Optimizing the use and energy consumption of on-premise hardware
2. Handprint (Sustainability by IT): it refers to the positive impact of using IT services and technology to help our customers reduce the impact on the environment of their services across all industries. Typical activities include:
    - Providing systems for Greenhouse gas emission reporting and visibility
    - Digitization of processes that would otherwise harm the environment
    - Providing technology to implement sustainable supply chains
    - Using technology to enable energy-efficient buildings
3. Heartprint: it refers to the positive impact of IT services on people, communities, and society. Typical activities include:
    - Social Responsibility programs
    - Employing women and minorities in IT
    - Sustainability hackathons

Also, several reporting and accounting frameworks have emerged for the topic of sustainability including ESG (Environment, Social, and Governance) factors, SROI (Social Return of Investment), Return of Sustainable Investment (ROSI), and so on. Companies are increasingly urged by various stakeholder groups, from customers to governments to society, to fulfill their sustainability responsibilities. According to [9], the 95% of the world's 250 largest companies already published a corporate sustainability report.

Sustainable management is understood as a long-term process of simultaneous optimization of these ESG performances, considering natural resource constraints, so as to enable sustained business activities without compromising the needs of future generations [10]. Sustainability is a wide-ranging concept, and in recent years it has particularly taken a turn in the scientific community and, especially, in the world of ICT. The idea of environmental sustainability is first and foremost grounded in management research, particularly in the Natural-Resource-Based View theory [11]. This theory identifies three objectives that must be considered by managers to promote the environmental sustainability of enterprises:

1. pollution prevention, achieved by minimizing waste and emissions;
2. product management, addressed by taking into account stakeholder demands and optimizing the product life cycle;
3. sustainable development, achieved by reducing the organization's environmental footprint and committing to a long-term sustainability vision.

For the realization of these three goals, IT and IS are of particular importance. As ESG reporting becomes main-stream, IT Leaders and CIOs are increasingly expected to provide accurate and transparent reporting across all industries, including the impact of their own IT departments.

Focusing on the GHG protocol as a standard for measurement of GHG emissions, it establishes comprehensive global standardized frameworks to measure and manage greenhouse gas emissions from private and public sector operations, value chains, and mitigation actions. It measures emissions across the value chain, categorized into three scopes as shown. The GHG protocol is used to report the total emissions of a business in terms of CO_2eq. GHG reduction targets are then set based on a baseline year. This can be of two types:

- Absolute targets reduce absolute emissions over time: for example, reduce CO_2 by 25% below 2020 levels by 2030;
- Intensity reduces the ratio of emissions relative to a business metric over time: for example, reduce CO_2 by 12% per service between 2023 and 2024.

Even though the pursuit of a methodology has been questioned in recent years [12–16], a very recent work [17] shed light on the need of introducing a greener set of principles in scientific research and computational science. Regarding sustainable software design, Robillard [18] is to date the most recent work about methods for energy modeling and to analyze and optimize the energy consumption of a given program. Instead, several are the works on the monitoring phase of software [19–23]. Within sustainable software development, Schubert et al. [24] defined a profiler that relates energy consumption to code locations, while other works [25, 26] estimate the carbon intensity of software with AI. Advancements have been made also in detecting data centers' energy consumptions [27, 28].

However, a comprehensive framework that encompasses all the phases of software engineering production is still lacking, and little work on this frontier has been made.

3 Methodology

The environmentally sustainable computing (ESC) framework is based on a holistic approach that covers all aspects of sustainable IT, including the design and development of new IT systems, the interventions in the ongoing operations of running systems, as well as the governance criteria of accounting and reporting and of sustainable attitudes. The ESC framework schema is synthesized in Fig. 1.

Within this framework, three principal intervention phases have been identified: new releases, running systems, and governance. In the following, we outline each area of intervention, where positive impact is driven by technology and where we believe efforts should be concentrated:

- New releases: it aims to create new assets for customers based on principles and best practices related to sustainability. It represents the most innovative side, as it pushes toward the development of new and previously non-existent technological solutions. It also pushes for the inclusion of green concepts right from the early stages of development. This same stream is divided between:
  - Design: the principles and best practices of sustainability and sustainable development must be followed and pursued starting from the design phase. Here we refer in particular to the design, for example, of interfaces that respect certain green software criteria aimed at reduced or minimal energy consumption.
  - Development: it is about adopting principles that promote the analysis and sustainable development of sustainable systems. This includes learning the theory of green software, the use of tools that support the development of sustainable code, and the search for less energy-intensive natural systems than others.
- Running systems: in this case, we refer to the evaluation, adaptation, modification, and reconstruction of pre-existing systems to produce new and more sustainable solutions than they were previously. At the basis of this stream is the concept of measurement: without measuring, it is neither possible to evaluate where we are nor to improve. What we want, therefore, is to recover sustainability values to push for even greater sustainability. In particular, we move between:
  - Monitoring: in this phase, methods are studied for monitoring the most relevant sustainability KPIs such as energy consumption and CO_2 emissions associated with the work carried out (or not) by customer projects, systems, and programs. It contains within it the search for methods to carry out monitoring, the search for the most suitable KPIs for an accurate evaluation of the system, and the data visualization useful for summarizing the situation more simply and intuitively.
  - Refactoring: this last phase can be understood as after the monitoring phase. Based on the measurements carried out, it is possible to identify possible problems and make changes to the systems to improve them. In particular, when it comes to sustainability, we push ourselves to build a system that can consume less energy, that is as reusable as possible, that produces the least amount of emissions and that respects the environment more.
- Governance: plays a crucial role in overseeing and steering sustainable computing practices. By addressing these governance aspects, organizations can holistically embed

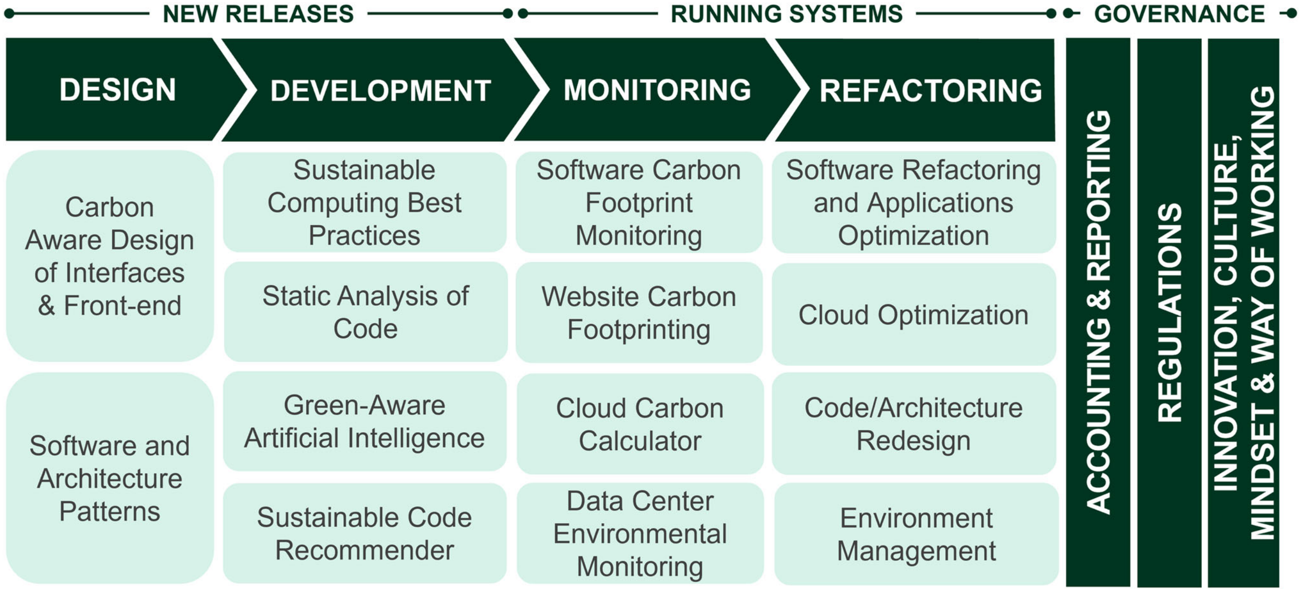


Fig. 1 The ESC framework is represented with horizontal dark green boxes representing the four software engineering phases of design, development, monitoring, and refactoring; while the horizontal light green boxes represent a particular intervention toward an ESC process; then, the vertical dark green boxes depict the governance attitudes that are intrinsic in every phase

sustainability into their computing practices, achieving long-term environmental benefits:

- Accounting and reporting: it involves the establishment of robust accounting and reporting mechanisms to track and communicate the environmental impact of computing practices. It acts as a transparency measure, providing stakeholders with clear insights into the sustainability performance of the organization's digital initiatives.
- Regulations: they constitute an important aspect of governance, ensuring that computing practices align with environmental standards and legal requirements. This involves adherence to existing regulations and proactive participation in the formulation of new ones that promote sustainability.
- Innovation, Mindset, and Way of Working: this aspect focuses on instilling a culture of innovation, environmental consciousness, and adaptive working practices within the organization. It recognizes that true sustainability goes beyond compliance, requiring a mindset shift and a commitment to continuous improvement.

3.1 Design

Environmentally sustainable design of algorithms, architectures, workflows, user interfaces (UIs), and user experiences (UXs) drive positive impacts from provisioning, use, and effective lifetimes of computing. Design choices may include also operating systems and middleware, or programming languages and compilation.

In general, as starting pillar of the ESC framework depicted in Fig. 1, green software design aims to produce novel approaches that address and increase the environmental sustainability of energy efficiency, performance, security, and other common problems.

3.1.1 Carbon-aware design of interfaces and front-end

In addressing the challenge of minimizing the Internet's carbon footprint, the escalating demand for data and the subsequent surge in energy consumption pose significant hurdles [29]. Websites and digital applications, in their pursuit of greater data intensity, demand increased bandwidth and processing power, leading to higher energy consumption and associated carbon emissions [30]. To mitigate this, the design of carbon-aware web interfaces and digital experiences emerges as a crucial strategy, fostering energy-efficient choices and optimizing resource utilization [31].

A key tactic involves nudging users toward low-data alternatives during periods of high carbon intensity in the energy grid, particularly when reliance on fossil fuels is pronounced. To ascertain the energy sources, interfaces can incorporate real-time indicators or information on the energy mix of the data centers serving the application. This transparency informs users about the percentage of energy derived from renewable or non-renewable sources. Also, interfaces could dynamically notify users during periods when the energy grid relies more heavily on fossil fuels. This awareness empowers users to make informed decisions about their interaction with the application. While data consumption can be task-dependent, interfaces can implement adaptive content-loading strategies. For resource-intensive tasks, such as video streaming, interfaces might offer lower-resolution options or suggest postponing non-urgent tasks to periods of higher renewable energy availability.

The lack of awareness regarding the carbon footprint associated with internet usage presents another challenge. Carbon-aware design interventions can address this by providing real-time information on the environmental impact of specific actions, fostering user consciousness about their digital activities' ecological consequences. Allowing users to set preferences for the data-performance trade-off gives them control over their environmental impact.

Opportunities for carbon footprint reduction lie in optimizing web pages and application efficiency. This involves minimizing unnecessary data transfers, file compression, and judicious use of energy-intensive features. The toolkit presented in Fig. 2 exemplifies an approach for designing sustainable digital products.

Additionally, carbon-aware design promotes sustainable user behavior by incorporating features such as reduced screen brightness or dark mode, effectively reducing energy consumption for devices with OLED or AMOLED screens. These efforts contribute to sustainable software design and foster environmentally conscious user behavior, collectively working toward the reduction of the environmental impact of internet-based applications and services.

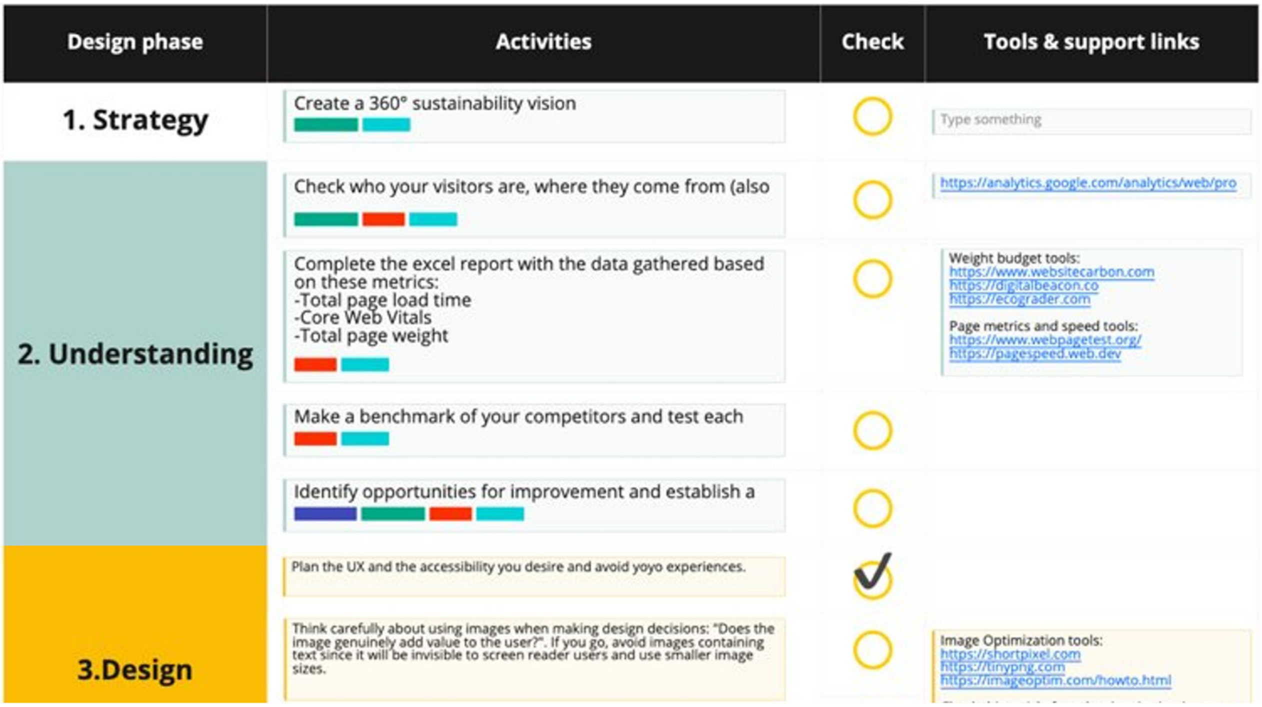
3.1.2 Software and architecture patterns

Software and architecture patterns play a pivotal role in sustainable software design, offering essential guidelines for creating energy- and resource-efficient systems [32]. The creation of carbon-aware design patterns necessitates a comprehensive consideration of various software development and architectural aspects while prioritizing sustainability [33, 34].

A primary objective of carbon-aware design patterns is the optimization of energy consumption. Achieving this involves the adoption of efficient algorithms, reduction of unnecessary computations, and minimizing power-intensive operations. Strategies such as employing caching techniques to reduce database queries or optimizing network communications contribute significantly to improving energy efficiency.

Optimizing resource usage, encompassing memory, processing power, and storage, is another critical aspect. Techniques like data compression, deduplication, and efficient memory management are instrumental in minimizing resource requirements, thereby reducing the overall energy consumption and carbon footprint of the software system.

Design patterns facilitating scalability and elasticity are integral to sustainable computing. Dynamic resource adjust-



Design phase	Activities	Check	Tools & support links
1. Strategy	Create a 360° sustainability vision 🟩 🟦	○	Type something
2. Understanding	Check who your visitors are, where they come from (also 🟩 🟥 🟦	○	https://analytics.google.com/analytics/web/pro
	Complete the excel report with the data gathered based on these metrics: -Total page load time -Core Web Vitals -Total page weight 🟥 🟦	○	Weight budget tools: https://www.websitecarbon.com https://digitalbeacon.co https://ecograder.com Page metrics and speed tools: https://www.webpagetest.org/ https://pagespeed.web.dev
	Make a benchmark of your competitors and test each 🟥 🟦	○	
	Identify opportunities for improvement and establish a 🟦 🟩 🟥 🟦	○	
3.Design	Plan the UX and the accessibility you desire and avoid yoyo experiences.	✓	
	Think carefully about using images when making design decisions: "Does the image genuinely add value to the user?". If you go, avoid images containing text since it will be invisible to screen reader users and use smaller image sizes.	○	Image Optimization tools: https://shortpixel.com https://tinypng.com https://imageoptim.com/howto.html

Fig. 2 A toolkit for designing sustainable digital products

ments based on demand, achieved through auto-scaling, load balancing, and resource pooling, prevent overprovisioning and enhance operational efficiency. Additionally, virtualization and containerization technologies contribute to carbon-aware design patterns by maximizing resource utilization.

Distributed computing patterns further optimize resource usage by distributing computational tasks across multiple nodes or data centers. This approach allows for load balancing, resource optimization, and the potential utilization of renewable energy sources in diverse geographical regions. In the context of energy efficiency, fault-tolerance strategies align with sustainable computing principles. These strategies optimize resource utilization by redistributing workloads to operational servers in the event of a server crash, preventing system outages and dynamically allocating resources based on actual demand. This dynamic resource allocation ensures servers operate closer to their capacity without wasting energy on idle resources.

Moreover, sustainable software design extends to user experience optimization. Intuitive and responsive user interfaces, coupled with features like energy-saving modes and personalized energy dashboards, encourage users to adopt sustainable practices, thereby contributing to the reduction of their carbon footprint. These multifaceted considerations collectively contribute to the development of sustainable software systems.

3.2 Development

The development sector of the ESC framework focuses on incorporating sustainable software development practices into the software development lifecycle. These practices, as shown in Fig. 1, include certification, static analysis of code, green-aware AI, and dynamic suggestions to developers. By incorporating these practices, developers can create software systems that are energy-efficient, resource-optimized, and aligned with environmental sustainability goals.

3.2.1 Sustainable computing best practices

The evolving intersection of technological advancement and environmental sustainability underscores the imperative for a comprehensive understanding and implementation of sustainable IT solutions [35]. The growing environmental impact of IT systems necessitates proactive measures to mitigate their effects, driving IT professionals and organizations to seek a deeper understanding of sustainability issues, enhance their sustainability practices, and adhere to environmental regulations [36].

Sustainable computing best practices serves as foundational guidelines throughout the software development lifecycle. These practices encompass diverse considerations such as energy efficiency, resource optimization, carbon footprint reduction, and eco-friendly design principles. Adherence to these practices is instrumental in realizing green software.

The adoption of specific principles further aids developers in integrating sustainability into their software projects. These principles include prioritizing functionalities with higher power consumption, minimizing data usage through efficient cache policies, enhancing energy efficiency by eliminating unused features, adapting computational accuracy based on operational needs, and real-time monitoring of energy consumption during software development. Additionally, the choice of programming language and AI model significantly impacts energy efficiency, warranting nuanced consideration. Developers are encouraged to stay informed about emerging technologies and actively participate in industry forums to contribute to the development of new standards promoting sustainable software development.

The GSF assumes a pivotal role in establishing and maintaining green IT practices. Through collaboration with industry experts, researchers, and practitioners, the GSF continuously evolves standards for sustainable software development. The GSF's certification, Green Software for Practitioners (LFC131),[1] offers a comprehensive curriculum, equipping participants with the knowledge to implement Sustainable IT practices. Figure 3 shows a certification diploma of the completion of the course.

Certification standards, exemplified by LFC131, provide a transparent framework for companies to measure and report their GHG emissions, fostering worldwide comparability and cost-effectiveness. Certification maintenance and updates are essential to ensure relevance and effectiveness. The GSF engages with practitioners, industry leaders, and experts to gather feedback, identify emerging challenges, and propose modifications or integrations to the certification framework. Developers can actively contribute to this iterative process, sharing insights and experiences to enhance sustainable software development practices.

The overarching goal is to devise proactive approaches for proposing new standards, maintaining and updating green IT practices, and suggesting modifications or integrations to the GSF's certification. By collating recognized practices and innovative solutions, a valuable learning path in sustainable IT can be defined, fostering a green-aware mindset among developers. Active participation in this collaborative effort empowers developers to shape the future of sustainable software development, making a meaningful contribution to mitigating the environmental impact of IT.

---

[1] https://training.linuxfoundation.org/training/green-software-for-practitioners-lfc131/.

THE LINUX FOUNDATION

Green Software Foundation

The Linux Foundation hereby certifies that

[redacted]

has successfully completed

Green Software for Practitioners (LFC131)

TIMOTHY SEREWICZ,
TRAINING PROGRAM DIRECTOR

[redacted]
DATE OF COMPLETION

[redacted]
CERTIFICATE ID NUMBER

Fig. 3 Linux Foundation and Green Software Foundation green software for practitioners (LFC131) certificate

3.2.2 Static analysis of code

As the IT industry expands, there's a heightened awareness of the environmental impact tied to software systems. Incorporating sustainable practices into the software development life-cycle is essential for constructing sustainable-by-design software [37]. Presently, these practices are not widespread due to inadequate definition, limited education for developers, and the subsequent under-utilization of associated benefits like code optimization and reduced energy costs.

Static code analysis, a process of scrutinizing source code without execution, proves instrumental in identifying potential issues, vulnerabilities, and performance bottlenecks [38]. In developing scoring models to track and minimize CO_2 emissions in software pipelines, static code analysis evaluates the energy efficiency and carbon footprint at the source code level. This allows developers to pinpoint areas for optimization to curtail energy consumption and CO_2 emissions.

Initiating the development of a scoring model involves establishing a baseline CO_2 emissions measurement, often done through green index calculations considering factors like energy consumption and resource usage. Figure 4 depicts the green coding pipeline involving static analysis, green index quantification, emissions evaluation, and code optimization.

Leveraging static analysis tools, developers can identify code patterns leading to increased energy consumption or CO_2 emissions. These might include inefficient algorithms, resource-intensive operations, or excessive data transfers. Static analysis flags these issues, offering developers insights for code optimization to reduce energy consumption.

Moreover, static analysis aids in informed decision-making regarding cloud infrastructure hosting. By discerning energy-intensive operations within the code, developers can opt for hosting in regions reliant on renewable energy sources, significantly diminishing the software pipeline's carbon footprint.

Scoring models developed through static analysis furnish actionable feedback and optimization suggestions to reduce CO_2 emissions. These models assign weights to different code components based on their energy consumption or car-

Fig. 4 Green coding pipeline: static analysis of code, Green index quantification, emissions evaluation and code optimization

bon emissions, allowing developers to prioritize optimization efforts effectively.

Static analysis's integration into the software development workflow, through IDE (Integrated Development Environment) integrations and code quality gates, delivers real-time feedback to developers. This early alert system notifies developers of potential energy inefficiencies or high carbon footprint code patterns as they write code, facilitating the early addressing of sustainability concerns and optimizing code to reduce emissions.

The role of static analysis in developing scoring models is pivotal for providing developers visibility into their code's environmental impact. By quantifying CO_2 emissions, developers can make informed decisions and undertake specific actions to actively diminish their software's carbon footprint.

3.2.3 Sustainable code recommender

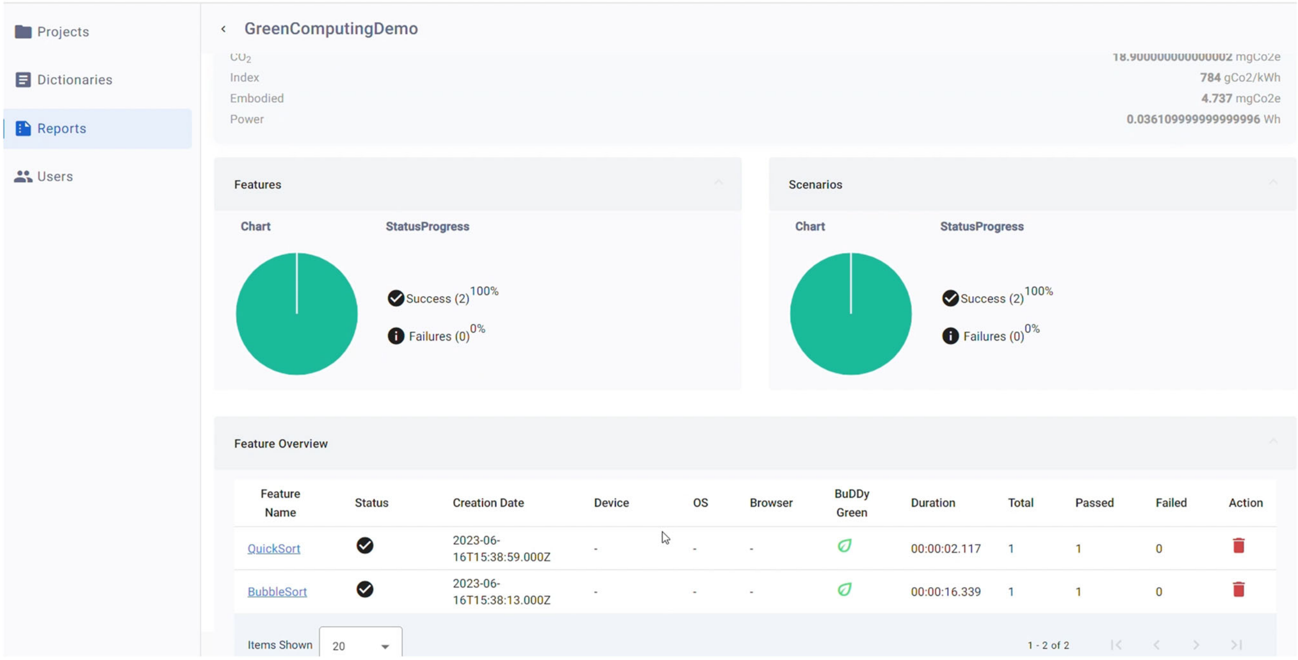
As environmental awareness rises, integrating sustainability practices into software development is paramount [39]. It is therefore also necessary to incorporate recommendations that encompass sustainability. The sustainable code recommender plays a pivotal role in promoting energy-efficient, resource-optimized, and environmentally friendly coding practices. Integrated into the IDE, this tool offers real-time recommendations, hints, and tips to developers during coding and maintenance, aligning with sustainable software development principles.

The recommender suggests adopting energy-efficient algorithms, optimizing data structures, and using caching techniques to reduce resource consumption, actively encouraging the adoption of sustainable practices. To enhance sustainability integration, an extension for carbon footprint calculation associated with test executions has been introduced, enabling developers to assess and minimize their code's environmental impact.

The recommender supports automatic tests, simulating different code paths to analyze energy consumption and resource utilization. This empowers developers to identify potential bottlenecks or energy-intensive operations and make informed decisions for code optimization. Figure 5 shows a tool that dynamically suggests a more sustainable piece of code or algorithm.

Additionally, it facilitates carbon-aware what-if scenarios, allowing developers to estimate the environmental impact of different configurations and infrastructure choices. Moreover, the recommender tracks and reports sustainability progress, providing metrics and insights on energy efficiency improvements over time. This aids developers in understanding the impact of their coding choices on the overall sustainability of the software solution.

Based on established best practices in optimized code development, the recommender employs quantitative metrics such as computational resource utilization, network bandwidth requirements, and storage occupancy. Automatic evaluation of energy efficiency and simulation of "what-if" scenarios enable the recommender to suggest optimizations, contributing to an overall reduction in the software's environmental impact. By monitoring tests, developers can identify



CO_2 18.900000000000002 mgCo2e
Index 784 gCo2/kWh
Embodied 4.737 mgCo2e
Power 0.036109999999999996 Wh

Feature Name	Status	Creation Date	Device	OS	Browser	BuDDy Green	Duration	Total	Passed	Failed	Action
QuickSort	✔	2023-06-16T15:38:59.000Z	-	-	-	🍃	00:00:02.117	1	1	0	🗑
BubbleSort	✔	2023-06-16T15:38:13.000Z	-	-	-	🍃	00:00:16.339	1	1	0	🗑

Fig. 5 Sustainable code recommender for energy-efficient and resource-optimized software development

low-emission test types, guiding their choice for more environmentally friendly options.

In essence, the sustainable code recommender, integrated into the development workflow, serves as an intelligent tool that not only guides developers toward sustainable coding practices but also actively contributes to the reduction of the environmental footprint of software applications.

3.2.4 Green-aware artificial intelligence

The exponential growth of AI, especially large-scale models like BERT [40], has significantly improved accuracy but also led to a surge in CO_2 emissions due to heightened electricity demands [41]. While AI's transformative applications continue to expand, there is a critical need to address the escalating energy consumption inherent in its development and usage [42, 43].

The power consumption of machine learning (ML) and deep learning (DL) models is particularly evident in two main phases: training and inference [44]. Training, which occurs once for a fixed amount of data, consumes considerably more power than inference. Future AI energy demand is expected to rise, necessitating efforts to curtail power consumption.

Three primary approaches aim to reduce AI power consumption:

1. Facility-level power reductions: Focusing on energy-efficient facilities for computational resources used in training and inference.
2. Device-level power optimization: Tailoring hardware choices for AI models based on characteristics and constraints, optimizing power use at the device and component levels.
3. Algorithmic power reduction: Developers implementing power-efficient algorithms, including model compression, suitable data formats, and energy-efficient programming languages.

Model compression involves three main techniques: pruning, distillation, and quantization, each involving a trade-off between power reduction and model accuracy. Pruning entails removing less important parameters [45], distillation

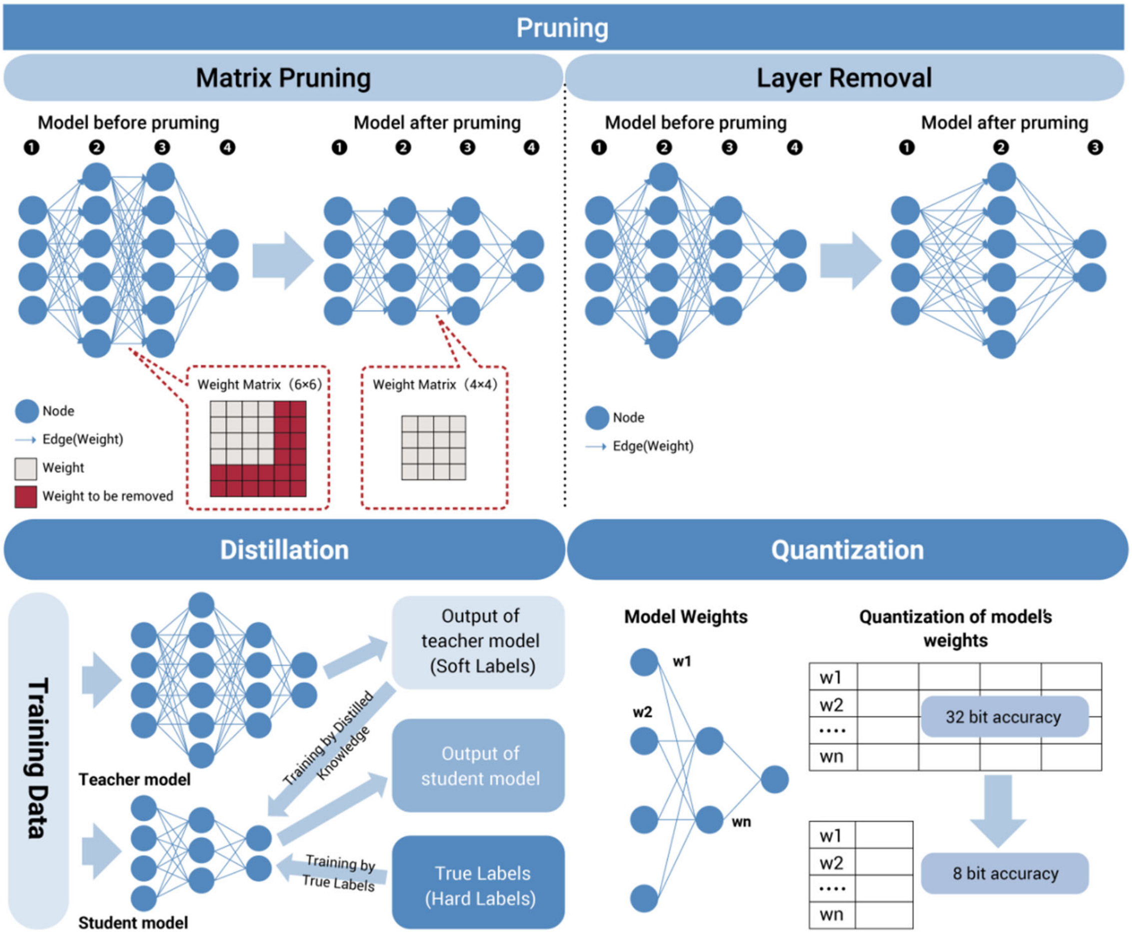


Fig. 6 AI model reduction techniques: pruning, distillation, quantization

involves a small model mimicking a larger one, and quantization reduces the precision of model weights without altering its structure. Figure 6 shows the three different approaches to reducing AI power consumption.

In the inference phase, green-aware AI offers optimization opportunities, considering factors like model selection, deployment strategies, and inference optimizations. Techniques such as model quantization and intelligent deployment strategies can significantly minimize computational requirements during inference, reducing energy consumption. Moreover, green-aware AI aids in identifying energy-intensive operations or code patterns within ML and DL tasks, allowing developers to focus on optimizing these areas for overall energy savings.

Additionally, green-aware AI facilitates data-driven decisions by predicting energy consumption or estimating the carbon footprint associated with ML or DL tasks. Leveraging historical data or AI models trained on energy consumption patterns empowers developers to proactively address and mitigate the environmental impact of their software.

Within the ESC framework, developers can employ green strategies to reduce energy consumption in both the training and inference phases. This includes model lightweighting, resource allocation optimization, deployment strategies, inference optimizations, and energy consumption predictions. These strategies align with sustainable software development practices and contribute to the principles of carbon-aware computing, ultimately reducing the environmental impact of ML and DL tasks.

3.3 Monitoring

In the context of the monitoring phase, the main idea is to endow the ESC framework (ref. Fig. 1) with tracking actual IT's energy and carbon footprint. The primary purpose is to help gain a comprehensive understanding of the environmental impact and identify areas where to reduce carbon footprint and optimize software and hardware usage. The monitoring activity can be identified in four fields: software, websites, cloud, and data centers.

3.3.1 Software carbon footprint monitoring

In the journey to reduce GHG emissions, the prominent need is to take software's carbon footprint into account and rethink the way software is designed, developed, and deployed [46]. Anyway, the monitoring solution is the first step in assessing the current IT Carbon Footprint [47, 48]. Given the customer's need, a system that is capable to calculate carbon emissions and analyze results on a perimeter of IT applications is demanded. In this context, the solution comes from the principles of the GSF and the SCI framework. In particular, the monitoring tool uses two main key performance indicators (KPIs): CO_2eq for overall carbon emissions and SCI for emission rates, as recommended by the GSF. By using these KPIs, the tool is able to measure software carbon emissions and energy consumption accurately and identify opportunities for improvement. Figure 7 shows an excerpt of the website of Electricity Maps[2] whose aim is to monitor carbon emissions and energy consumption. These data can be then gathered to calculate the Software Carbon Intensity.

Hence, software carbon footprint monitoring allows the inspection of the electricity consumption and emissions of IT systems in order to carry out targeted redesign or refactoring actions and measure the results. After defining the data model and data collection method, a variety of emission factors from a range of validated sources can be used to calculate the CO_2eq values. The resulting KPIs can be then prepared and organized to populate a dashboard for better insights analysis and coherent data visualization. Using these solutions, stakeholders may promote asks to compute the amount of GHG emissions produced by IT applications (CO_2eq), and the SCI to monitor the efficiency of software and identify areas for improvement. By reducing the SCI, we can make a significant impact on reducing our carbon footprint and promoting sustainability.

3.3.2 Website carbon footprinting

Website carbon footprinting tools and methodologies use data on energy sources used by data centers, the energy efficiency of servers, network performance, and user device power consumption to make accurate calculations. This information enables developers and businesses to gain a comprehensive understanding of the environmental impact of their websites [49, 50].

The role of website carbon footprinting goes beyond mere data calculation. It serves as a powerful awareness-raising tool, inspiring a commitment to sustainability in the digital landscape. By making website owners and users aware of their carbon emissions, it fosters a sense of responsibility toward reducing the environmental impact of online activities.

Armed with insights from website carbon footprinting, developers can implement sustainable design principles to optimize website performance and reduce energy consumption. This includes practices such as efficient coding, using compression techniques to reduce file sizes, and leveraging caching to minimize server requests. These design principles not only improve the website's environmental impact but also enhance the user experience by ensuring faster load times and smoother browsing.

Additionally, website carbon footprinting provides guidance on reducing data transfer requirements. This can be

---

[2] https://app.electricitymaps.com/.

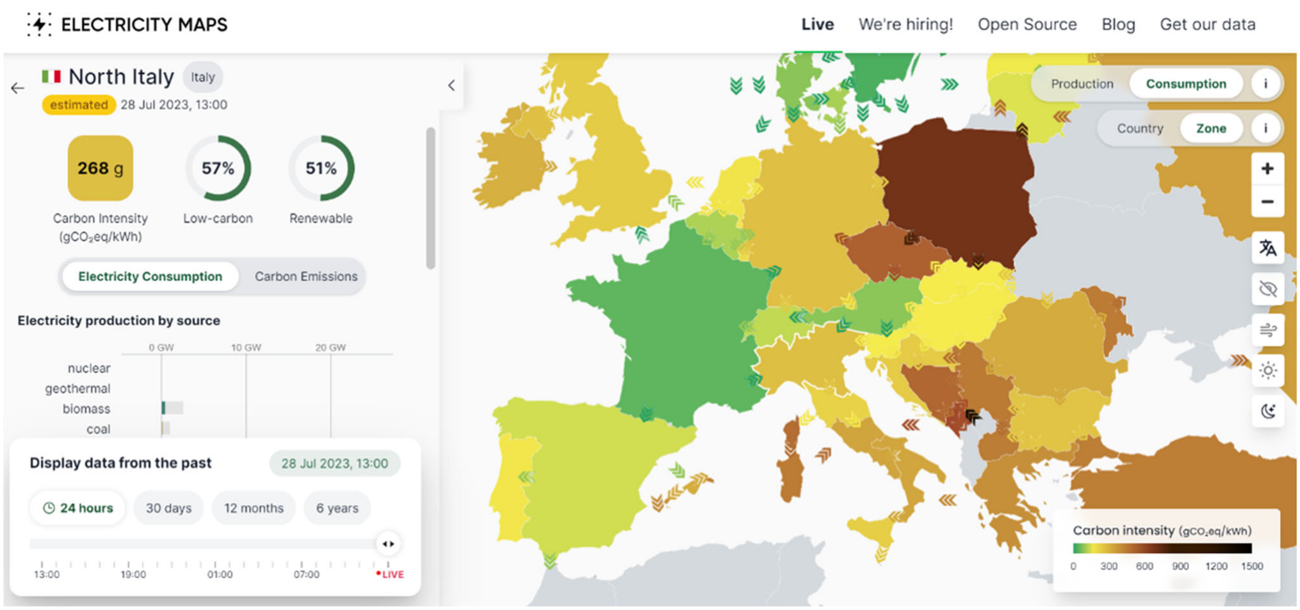


Fig. 7 Example of carbon intensity and electricity mix mapping from electricity maps

achieved through strategies like lazy loading of images, optimizing video content, and adopting efficient data formats. By reducing unnecessary data transfer, the carbon emissions associated with website usage can be significantly decreased.

Moreover, website carbon footprinting encourages responsible user behaviors. By providing feedback on the environmental impact of different online activities, it empowers users to make conscious choices. For example, users can be nudged to reduce video streaming quality when their internet connection relies heavily on fossil fuels or to enable energy-saving modes on their devices during prolonged browsing sessions.

3.3.3 Cloud carbon calculator

The cloud carbon calculator acts as an emissions measurement and analysis tool that utilizes best practice methodologies to estimate the energy usage and carbon emissions of cloud resources [51]. By collecting usage data from major cloud providers, the calculator can calculate the estimated energy consumption in Watt-Hours and convert it into greenhouse gas emissions in metric tons of CO_2eq.

This tool provides organizations with valuable metrics and carbon savings estimates that can be shared with employees, investors, and other stakeholders. By transparently disclosing the carbon footprint of their cloud operations, organizations can demonstrate their commitment to sustainability and promote carbon-aware computing practices.

Moreover, the cloud carbon calculator allows organizations to monitor and track their carbon emissions over time. This enables them to set targets for reducing their carbon footprint and measure the effectiveness of their sustainability initiatives. By providing continuous visibility into cloud-related carbon emissions, the calculator helps organizations make data-driven decisions to minimize their environmental impact.

The cloud carbon calculator's ability to analyze and track the carbon footprint of cloud resources also encourages responsible cloud resource management. By identifying energy-intensive or carbon-intensive operations within a cloud infrastructure, organizations can optimize their resource allocation and configuration to minimize energy consumption and carbon emissions. Figure 8 shows a tool that monitors cloud carbon footprint and provides cloud usage information.

Furthermore, the wide visibility across multiple cloud providers offered by the cloud carbon calculator is instrumental in fostering a culture of sustainability and accountability. It enables organizations to benchmark their carbon performance against industry standards and encourages healthy competition among cloud providers to offer more energy-efficient and carbon-aware services.

Additionally, the cloud carbon calculator plays a vital role in aligning organizations with environmental regulations and sustainability goals. By accurately measuring and reporting carbon emissions from cloud resources, organizations can ensure compliance with emission reduction targets, support broader sustainability initiatives, and make informed decisions about their cloud infrastructure by considering the environmental impact and sustainability performance of different cloud service providers.

Cloud Providers: 3 of 3

Accounts: 15 of 15

Services: 7 of 7

Your cumulative emissions are
275 metric tons CO2e
that is equivalent to

Rows per page: 25

Potential Carbon Savings (t)
11.492
11.984
1.288
1.892
11.972
1.466
1.803
2.855
0.168
2.875
14.929
< 0.001
0
0.001
0.002

Rows per page: 25

Fig. 8 Cloud carbon footprint monitoring dashboard

3.3.4 Data center environmental monitoring

Data center environmental monitoring is pivotal for evaluating and refining the ecological footprint of data center operations [27, 52–54]. Analyzing KPIs provides organizations with insights to make data-driven decisions, enhancing energy efficiency and mitigating carbon emissions. Relevant KPIs include power usage effectiveness (PUE), carbon usage effectiveness (CUE), energy efficiency index (EEI), and water usage effectiveness (WUE), offering a comprehensive view of the data center's environmental performance.

Efficient monitoring involves the creation of a Digital Twin, a synchronized virtual replica of the actual data center. This Digital Twin, dynamically linked to the data center's monitoring systems, continuously updates with real-time data, providing stakeholders with current insights into environmental performance. To ensure user comprehension, interactive dashboards, visualizations, and charts are employed. Color-coded indicators or animations highlight critical aspects, aiding quick identification of potential issues. Figure 9 shows a tool that monitors a data center with color-coded indicators and animations.

The integration of data center environmental monitoring into the project framework necessitates technological, graphical, and data integration. Technologically, data center monitoring systems and sensors must connect to the Digital Twin for real-time updates. Graphically, the visualization technique should seamlessly integrate into the project's user interface, catering to various stakeholders for collaborative decision-making. From a data standpoint, the monitoring system should aggregate relevant data from various sources within the data center, including power consumption, cooling system performance, and renewable energy usage.

This monitoring approach contributes to sustainable software practices and carbon-aware computing. By tracking and optimizing energy and resource consumption, organizations can identify areas for improvement and reduce their data center's carbon footprint, fostering overall sustainability in data center operations.

3.4 Refactoring

Refactoring practices are addressed to existing software and infrastructure that are not optimized for low carbon emission. Following the dedicated pillar in Fig. 1, these practices include software refactoring and application optimization, cloud optimization, code/architecture redesign, and environment management.

3.4.1 Software refactoring and applications optimization

Software refactoring involves restructuring code to enhance quality, maintainability, and efficiency without altering functionality. Sustainable software refactoring extends this practice by integrating green software engineering principles [55–58], focusing on identifying and addressing energy-intensive or resource-heavy code patterns. The goal is to reduce energy consumption during software execution, contributing to carbon-aware computing.

In parallel, sustainable application optimization fine-tunes software performance and resource utilization, considering the environmental impact of these optimizations. This

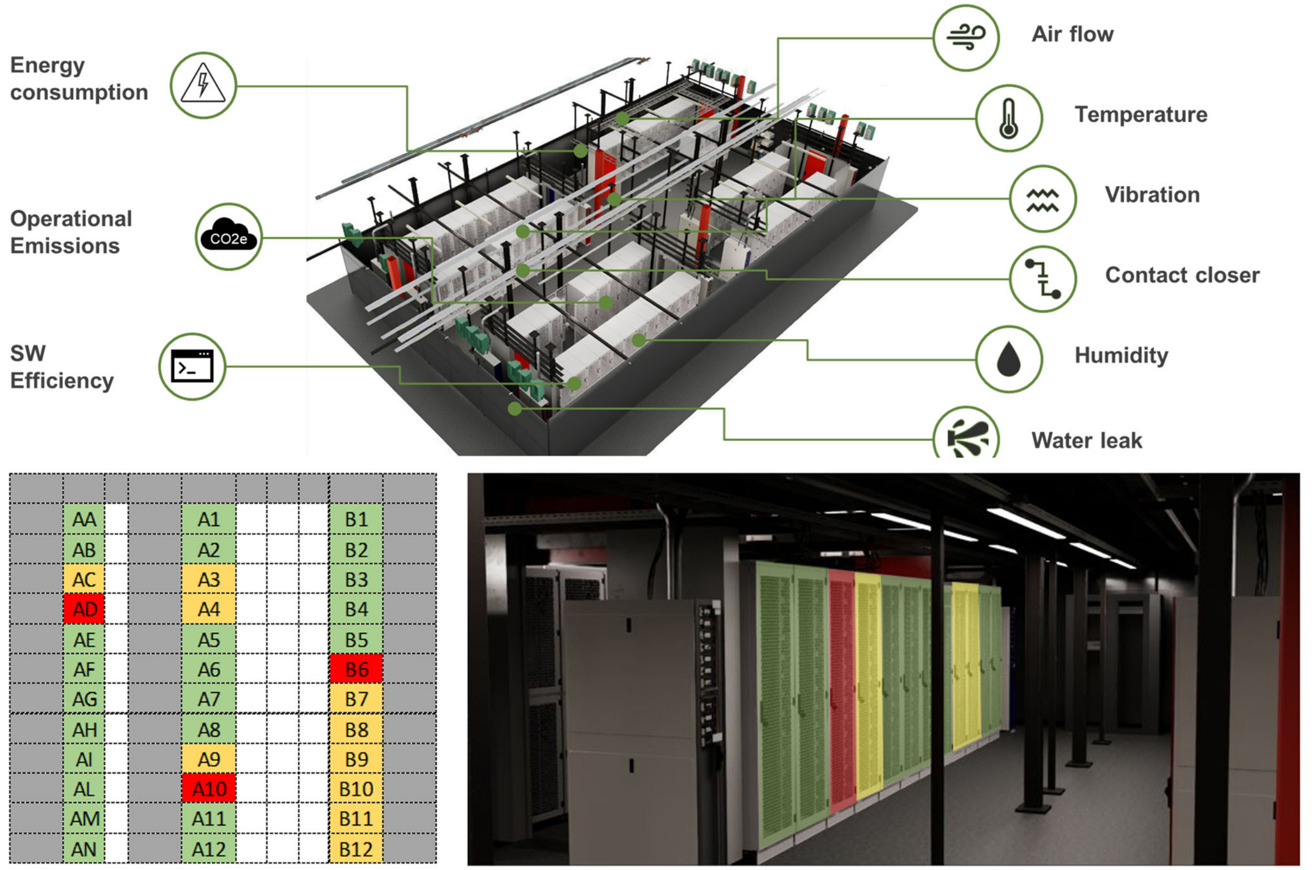


Fig. 9 Data center digital twin environmental monitoring

involves tasks such as optimizing database queries, reducing memory usage, and enhancing application responsiveness. The collective impact is a reduction in the energy consumption and carbon footprint associated with software operation.

Sustainable software refactoring and application optimization strategies can be designed and developed with several key principles in mind:

- Energy-conscious code review: Developers can actively review code to identify energy-intensive operations and suggest energy-efficient alternatives. This code review process includes identifying memory leaks, reducing nested loops, and optimizing I/O operations.
- Algorithm optimization: By optimizing algorithms to use fewer computational resources, developers can significantly impact energy efficiency. Choosing algorithms with lower time complexity and reducing unnecessary iterations can lead to considerable energy savings.
- Resource utilization monitoring: Implementing monitoring mechanisms to track resource utilization during application execution allows developers to identify bottlenecks and opportunities for optimization. Real-time resource monitoring helps detect excessive CPU usage, memory consumption, or disk I/O, leading to more efficient resource management.
- Green design patterns: Incorporating green design patterns into the refactoring and optimization process can further enhance sustainability. For instance, using lazy loading techniques, caching, and lightweight data structures can improve application performance and reduce energy consumption.
- Performance benchmarking: Measuring and benchmarking the energy consumption and performance of different application versions can aid in identifying the most energy-efficient implementation. Performance benchmarking helps developers make data-driven decisions about the effectiveness of refactoring and optimization efforts.

By integrating these principles into the software development workflow, organizations foster a culture of sustainable software development. This approach leads to energy-efficient and environmentally friendly software solutions, aligning with green software engineering principles and contributing to carbon-aware computing.

3.4.2 Cloud optimization

Sustainable software refactoring combined with cloud optimization allows developers to create energy-efficient and environmentally friendly software applications that run on cloud platforms [59]. Cloud optimization involves fine-tuning the utilization of cloud resources to maximize efficiency and minimize waste [60].

One of the key aspects of cloud optimization is adopting location-shifting strategies. Location shifting involves choosing data centers or cloud regions that rely on renewable energy sources or have lower carbon intensity. By strategically selecting cloud regions based on their energy profiles, developers can reduce the carbon footprint of their applications. Cloud providers often offer a variety of data centers located in different regions worldwide. Sustainable software refactoring can enable applications to take advantage of these options and run in regions with abundant renewable energy resources, reducing their overall carbon impact.

Time shifting is another important approach in cloud optimization. By scheduling compute-intensive tasks during off-peak hours or times when the energy grid relies more on renewable energy sources, developers can minimize the carbon emissions associated with cloud resource usage. Sustainable software refactoring can include optimizing task schedules and implementing workload-balancing algorithms that consider the environmental impact. By aligning resource-intensive tasks with renewable energy availability, developers can make their applications more energy-efficient and sustainable.

Demand shaping is yet another aspect of cloud optimization that can be combined with sustainable software refactoring. It involves adjusting resource allocation based on demand patterns to optimize efficiency. By dynamically scaling resources up or down as needed, developers can avoid unnecessary energy consumption during periods of low demand. This elastic scaling approach ensures that resources are utilized efficiently, leading to reduced carbon emissions and optimized energy consumption.

Cloud optimization also involves adopting containerization and serverless computing approaches. Containers allow developers to package applications and their dependencies, enabling seamless deployment and resource utilization. Serverless computing, on the other hand, abstracts server management, automatically adjusting resources based on demand and reducing idle energy consumption.

By refactoring applications to utilize containerization and serverless computing, developers can further optimize energy consumption and resource utilization. These modern approaches facilitate efficient cloud deployment and contribute to sustainable software development. Additionally, developers can incorporate cloud cost optimization practices. By continuously monitoring resource usage and identifying cost-saving opportunities, organizations can ensure that cloud resources are utilized efficiently, leading to financial savings as well as reduced environmental impact.

3.4.3 Code/architecture redesign

Code and architecture redesign involve restructuring and improving the codebase and system architecture to enhance efficiency, maintainability, and performance [61]. Sustainable software refactoring goes beyond traditional refactoring goals and focuses on identifying and addressing energy-intensive or resource-heavy code patterns to reduce energy consumption and carbon emissions during software execution.

One of the key aspects of code and architecture redesign for sustainable software development is the exploration and implementation of green architecture patterns. Green architecture patterns are design templates or solutions that promote energy efficiency, resource optimization, and environmental sustainability. These patterns may include architectural decisions such as using microservices, adopting event-driven architectures, or implementing distributed caching. By refactoring the code and architecture to incorporate these patterns, developers can optimize energy consumption and improve the overall sustainability of the software.

Sustainable software refactoring also involves the identification and replacement of energy-inefficient algorithms or operations. By analyzing the codebase and profiling the application's energy usage, developers can target areas that contribute most to energy consumption. For example, inefficient sorting algorithms or resource-intensive data processing can be refactored to more energy-efficient alternatives. This code-level optimization directly contributes to carbon-aware computing, reducing the software's carbon footprint.

In addition to code-level refactoring, architectural choices can significantly impact energy consumption. By redesigning the system architecture to support better resource utilization and scalability, developers can create energy-efficient applications. For instance, using cloud-native architectures that automatically scale resources based on demand can optimize energy consumption and reduce idle resource usage. Similarly, adopting containerization and serverless computing approaches can lead to more efficient resource allocation and energy usage.

Sustainable software refactoring practices also involve extensive testing and benchmarking to validate the effectiveness of the implemented changes. Developers conduct energy profiling and performance tests to measure the impact of code and architecture redesign on energy consumption. By analyzing the results, developers can fine-tune the refactoring efforts and ensure that the software meets the intended energy efficiency and sustainability goals. Continuous monitoring and analysis of the application's energy consumption provide valuable insights into its carbon footprint and help guide further refactoring and optimization.

Moreover, through code and architecture redesign, developers contribute to the creation of new literature on green computing approaches to software development. Sharing the experiences and lessons learned from sustainable software refactoring projects provides valuable insights for the broader software development community. By documenting successful green architecture patterns, optimization strategies, and energy-efficient design decisions, developers can inspire and guide others in adopting sustainable software development practices.

3.4.4 Environment management

By introducing green approaches to industrial software deployment stages: development, test, and production, organizations can optimize energy consumption and reduce the carbon footprint of their software solutions [55, 57, 61, 62]. Indeed, environment management involves the efficient allocation and utilization of resources at each stage of the software development process. This includes managing development environments, testing environments, and production environments to ensure they align with green computing principles.

In the development stage, environment management focuses on providing developers with tools and resources that promote sustainable software refactoring practices. Development environments should support energy profiling and analysis tools, allowing developers to identify energy-intensive code segments and inefficient algorithms. By incorporating energy consumption metrics into the development environment, developers gain insights into the environmental impact of their code changes and can actively work toward reducing energy consumption and carbon emissions. Furthermore, environment management at the development stage should encourage collaboration and knowledge sharing around sustainable software refactoring practices. This can be achieved through workshops, training sessions, and sharing best practices among development teams.

In the test stage, environment management ensures that testing environments mimic real-world production scenarios and accurately represent the energy consumption patterns of the software in a production setting. By using realistic data and traffic patterns, testing environments can provide accurate energy consumption estimates during stress testing and performance evaluation. This allows organizations to make data-driven decisions about the energy efficiency of their software before it is deployed to production. Additionally, environment management should enable energy efficiency testing as part of the testing process. Energy profiling tools and benchmarks can be integrated into the testing environments, enabling organizations to measure and optimize the energy consumption of their software during various test scenarios.

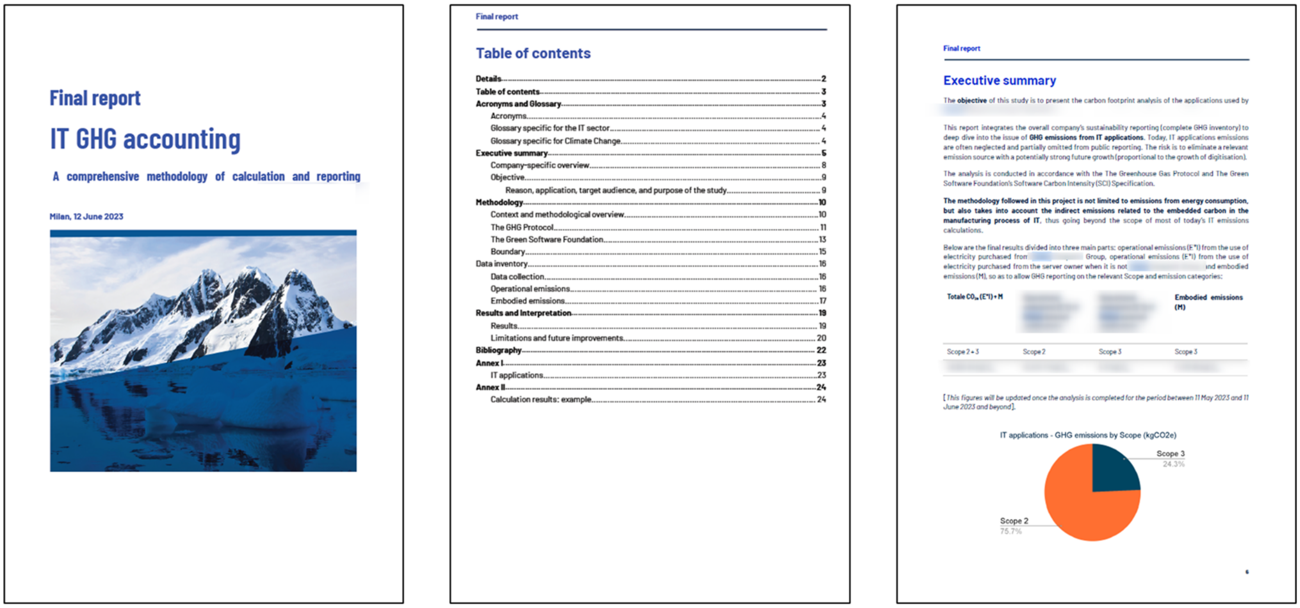
In the production stage, environment management focuses on optimizing the deployment environment to minimize energy consumption and carbon emissions. This involves using green cloud infrastructure, data centers powered by renewable energy, and energy-efficient server configurations. By strategically choosing data centers or cloud regions with low carbon intensity, organizations can ensure that their production environment aligns with carbon-aware computing principles. Environment management in the production stage also includes monitoring and optimizing the energy usage of the deployed software in real time. Continuous monitoring and analysis of energy consumption metrics allow organizations to identify opportunities for further sustainable software refactoring and performance optimization. Moreover, energy-aware load balancing and dynamic resource allocation can be implemented in the production environment to ensure that resources are used efficiently and wastage is minimized.

3.5 Accounting and reporting

As organizations increasingly acknowledge their environmental responsibilities, the need to accurately measure, track, and communicate their environmental impacts becomes paramount. Accounting for carbon emissions, energy consumption, and other sustainability metrics enables organizations to gain a comprehensive understanding of their IT footprint, identify areas for improvement, and establish measurable goals.

As a cross-section of the ESC framework depicted in Fig. 1, accounting for IT sustainability metrics facilitates compliance with environmental regulations and standards. As governments worldwide intensify their focus on carbon reduction and sustainable development, organizations must demonstrate adherence to regulatory requirements. Robust accounting practices ensure that organizations can accurately report their environmental performance, enhancing their reputation and credibility in a rapidly evolving business landscape.

Transparent reporting of IT sustainability metrics enables stakeholders, including employees, shareholders, and customers, to understand the organization's commitment to sustainability. This transparency builds trust and confidence among stakeholders, contributing to greater loyalty and brand value. A comprehensive approach is required to enable organizations to have a holistic view of their indirect carbon footprint, encompassing the full range of energy-related activities as well as embodied emissions. As a result, the report becomes a strategic asset for assessing the true extent of the organization's environmental impact and identifying



Final report

IT GHG accounting

A comprehensive methodology of calculation and reporting

Milan, 12 June 2023

Final report

Table of contents

Final report

Executive summary

The **objective** of this study is to present the carbon footprint analysis of the applications used by [redacted]

This report integrates the overall company's sustainability reporting (complete GHG inventory) to deep dive into the issue of **GHG emissions from IT applications**. Today, IT applications emissions are often neglected and partially omitted from public reporting. The risk is to eliminate a relevant emission source with a potentially strong future growth (proportional to the growth of digitisation).

The analysis is conducted in accordance with the The Greenhouse Gas Protocol and The Green Software Foundation's Software Carbon Intensity (SCI) Specification.

The methodology followed in this project is not limited to emissions from energy consumption, but also takes into account the indirect emissions related to the embedded carbon in the manufacturing process of IT, thus going beyond the scope of most of today's IT emissions calculations.

Below are the final results divided into three main parts: operational emissions (E*I) from the use of electricity purchased from [redacted] Group, operational emissions (E*I) from the use of electricity purchased from the server owner when it is not [redacted] and embodied emissions (M), so as to allow GHG reporting on the relevant Scope and emission categories:

Totale CO_{2e} (E*I) + M	[redacted]	[redacted]	**Embodied emissions (M)**
Scope 2 + 3	Scope 2	Scope 3	Scope 3
[redacted]	[redacted]	[redacted]	[redacted]

[*This figures will be updated once the analysis is completed for the period between 11 May 2023 and 11 June 2023 and beyond*].

IT applications - GHG emissions by Scope (kgCO2e)

6

Fig. 10 Example of IT GHG accounting report

areas where energy efficiency measures can be implemented to minimize emissions. Figure 10 shows an example of IT GHG Reporting.

The sustainability report, empowered by robust accounting and adherence to standards, such as ISO14064, serves as a valuable benchmarking tool for the organization itself. By establishing a baseline of current emissions, the report becomes the foundation for setting ambitious yet attainable reduction targets. Regularly comparing emissions data against this baseline provides valuable insights into the organization's progress over time, motivating teams to continually seek innovative and sustainable solutions.

Beyond internal benchmarking, the report assumes a critical role in external communications. As organizations increasingly embrace corporate sustainability as a core value, stakeholders, including customers, investors, and regulatory bodies, demand greater transparency on environmental performance. The comprehensive sustainability report, underpinned by sound accounting practices, builds trust and fosters credibility among stakeholders, enhancing the organization's reputation as a responsible corporate citizen.

3.6 Regulations

In the spirit of the ESC framework depicted in Fig. 1, regulations pertaining to environmentally sustainable computing play a pivotal role in mitigating the adverse impact of the digital industry on our environment. These regulations encompass a multitude of facets within the realm of computing, ranging from device manufacturing to the management of electronic waste. By adhering to stringent guidelines and adopting carbon-aware computing practices, the digital industry can significantly minimize its carbon footprint and contribute to a Sustainable IT ecosystem.

Encouraging energy-efficient designs and practices is imperative in minimizing energy consumption within the digital industry. Governments and regulatory bodies often establish energy consumption standards for computers and other electronic devices, compelling manufacturers to develop products that operate with reduced power requirements. Additionally, promoting the use of renewable energy sources, such as solar and wind power, in data centers and server farms contributes to Sustainable IT. By harnessing clean energy alternatives, the environmental impact arising from reliance on fossil fuels can be mitigated, resulting in a substantial reduction in greenhouse gas emissions.

E-waste management regulations are equally critical in fostering environmentally sustainable computing practices. The rapid proliferation of electronic devices has led to a significant increase in electronic waste, necessitating appropriate guidelines for its handling and disposal. E-waste consists of hazardous materials like heavy metals, flame retardants, and other toxic substances that pose a serious threat if irresponsibly discarded. Implementing effective e-waste management systems entails recycling electronic waste through proper channels, thereby mitigating the release of hazardous substances into the environment. Moreover, it encompasses responsible disposal methods that prevent e-waste from ending up in landfills, where it can contaminate surrounding soil and water sources.

To ensure the efficacy of these regulations, industry stakeholders, governments, and regulatory bodies must collaborate to foster a supportive framework conducive to sustainable IT practices. Encouraging research and development in green technologies, such as energy-efficient components and recyclable materials, can drive innovation and facilitate the adoption of environmentally friendly solutions. Furthermore, effective monitoring and enforcement of regulations can promote industry compliance and ensure that sustainable practices are followed throughout the lifecycle of computing devices. Given these developments, green software is expected to become increasingly prevalent in global legislative, regulatory, and reporting domains over the next decade.

3.7 Innovation, mindset and way of working

Integrating sustainability into the corporate culture also enhances employee engagement and satisfaction, as motivated by the ESC framework depicted in Fig. 1. Many employees are increasingly concerned about environmental issues, and working for a socially responsible and sustainable company can improve their morale and commitment to the organization [63]. When companies invest in research and development of sustainable IT, they are also investing in updating the IT infrastructure, modern and energy-efficient hardware, recyclable components, and low-carbon software solutions. Sustainable IT also promotes the adoption of circular economy principles, encouraging innovation in product design and business models to maximize resource reuse, minimize waste, and extend the lifespan of IT equipment [64, 65].

Therefore, embracing IT sustainability nurtures a culture of environmental consciousness within the organization. It highlights the importance of reducing the ecological footprint and promotes responsible resource management. By integrating IT sustainability into their operations, companies can position themselves as responsible global citizens, attract environmentally conscious talent, and drive positive change within their industries. The application of sustainable IT practices facilitates transparency by making complex sustainability data accessible and understandable. Transparent reporting encourages open communication and collaboration across departments, promoting cross-functional efforts toward sustainability goals. The monitoring tools encourage a culture of continuous improvement. By regularly assessing and revisiting sustainability scenarios, employees and teams seek ways to refine strategies, optimize outcomes, and continuously strive for higher environmental performance. This culture of improvement creates a dynamic and adaptive approach to sustainability, as employees embrace the idea that sustainability is an ongoing journey.

4 Use cases

This section focuses on two compelling use cases that showcase the practical implementation of sustainable IT within the banking and the multi-utility industries. By examining these specific scenarios, we aim to demonstrate how sustainable IT practices can effectively address environmental and social challenges while delivering positive outcomes. Through the analysis of these use cases, we seek to provide valuable insights into the benefits, limitations, and implications of adopting sustainable IT practices of the ESC framework in real-world contexts. The exploration of these use cases will also help elucidate the various factors that influence the successful implementation of sustainable IT projects, including technological considerations, organizational dynamics, and societal impacts. By examining the step-by-step process undertaken in these cases, readers will gain valuable insights into the implementation challenges faced, as well as the strategies adopted to overcome them.

4.1 Measuring bank's software carbon footprint and carbon intensity

In this use case, we delve into the Sustainable IT initiatives undertaken by one of the largest banking institutions in Italy to address the environmental impact of their IT systems. Recognizing the significant role of technology in their operations and the potential ecological consequences, the bank embarked on a comprehensive sustainability strategy that specifically targeted its IT infrastructure. The bank acknowledged that traditional IT practices, such as energy consumption and carbon emissions, contribute to a large part of the overall environmental footprint of the banking industry. However, tracking these metrics across a large and complex IT infrastructure presented a significant challenge.

The bank's IT system managers understood the need to address the impact of IT systems on the environment, and their search for solutions focused on finding ways to reduce this impact while still meeting business needs. With a clear recognition of the importance of complying with the business plan, which included specific ESG goals, they sought to align the function of IT with the overall business strategy and to identify opportunities for IT to contribute to the achievement of ESG objectives. This required a comprehensive understanding of the impact of IT systems on the environment and a commitment to finding ways to reduce this impact while still maintaining effective and efficient business operations.

In the banking industry, where IT systems are extensive and multifaceted, conducting a thorough assessment of environmental metrics becomes a daunting task. The challenge lies in aggregating data from diverse sources, including servers, applications, and network components, to gain a comprehensive understanding of the overall environmental impact. In the ESC framework vision, a more efficient IT is achieved through an iterative methodology based on the four steps of Design, Development, Monitoring, and Refactoring through improvement strategies. The four steps are replicated until the desired level of performance is achieved.

One challenge is introducing a methodology for developing new software that is designed sustainably. To tackle this problem, the bank introduced software sustainability in the development phase, introducing the Green Index, adopted from the Static Analysis of Code of the ESC framework, which is a measure of code sustainability that helps ensure software products are environmentally friendly and efficient. This indicator is used to measure how sustainable bank developers (as well as partners and suppliers) are in writing code. Shifting toward sustainable software development requires a cultural change within the organization. Developers may initially resist integrating environmental considerations into their coding practices. Additionally, incorporating the Green Index during the development phase posed challenges in terms of aligning it with existing workflows and ensuring developers embraced it as a valuable metric.

The score is based on static analysis of code, which is performed using SonarQube,[3] a widely used code quality management tool that helps to continuously inspect code for bugs, vulnerabilities, and code smells. The rules used to calculate the Green Index are derived from the analysis of standards provided by the Consortium for Information and Software Quality (CISQ). The CISQ standards cover various aspects of software quality, such as security, maintainability, reliability, and efficiency, and are widely recognized and adopted in the software industry. By leveraging the power of SonarQube and the expertise of CISQ, software products can be not only functional and reliable but also environmentally friendly and efficient: in fact, with the Automated Source Code CISQ Performance Efficiency Measure (ASCPEM) specification, we can establish a standard measure of performance efficiency by detecting violations of good architectural and coding practices that may result in inefficient operation, such as performance degradation or excessive use of processor resources.

The second challenge was to implement a reliable and accurate monitoring tool to calculate and track carbon emissions to calculate emissions at different granularity levels, identify anomalies, and prioritize areas for improvement. The first focus was to target the most impactful application regarding sustainability metrics. However, the ultimate goal was to extend the solution to the entire IT infrastructure, which presents a significant challenge given the size and complexity of the bank's IT systems. To tackle this challenge, a step-by-step approach was used, which involved identifying and prioritizing the most critical areas for improvement, implementing targeted solutions, and continuously monitoring and optimizing performance. The two main KPIs of CO_2eq and SCI have been identified for the case by leveraging the Software Carbon Footprint monitoring tool of the ESC framework. To track the CO_2 emissions of an IT system, it is necessary to calculate the total amount of energy consumed by the server and then multiply that figure by the amount of CO_2 emitted to produce that energy. The energy consumption of a server can be broken down into several major components, including the central processing unit (CPU), memory, and storage. A monitoring tool was implemented to measure the amount of CO_2 emitted depending on resource use.

The results of the monitoring tool were made available in an internal dashboard to provide real-time insights to the decision-makers. Through the dashboard, users can easily access emissions data for different periods and filter the data by services, offices, applications, and environments. The dashboard provides real-time values for the current SCI, for CO_2eq emissions in kilograms, as well as a percentage breakdown of emissions by resource usage. Additionally, the dashboard highlights areas where resources are consuming and producing the most emissions, allowing for targeted optimization efforts. The dashboard is depicted in Fig. 11.

Upon completion of static analysis to assess the codebase against CISQ guidelines and assign the Green Index, the application was deployed. Subsequently, the team employed the CO_2eq and SCI calculator to quantify the carbon footprint and the SCI of the application in a production environment. Daily calculations were conducted over a one-month period using real-time carbon intensity data. This daily approach aimed to enhance accuracy by avoiding the potential underestimation of emissions associated with the utilization of average values over extended periods. However, for reference, an average carbon intensity over the period was reported as 296.96 gCO_2eq/kWh. Monitoring encompassed a total of 76 software applications within the study's scope. Daily data on the total CO_2eq and SCI emissions for each application were collected, facilitating the aggregation of data for the entire month to derive the total CO_2 emissions for each software application and the collective emissions for all applications.

For example, the most impactful software application exhibited a consumption of $5,721.48$ $kgCO_2$eq in one month. Cumulatively, the total monthly emissions for all 76 software

---

[3] https://www.sonarsource.com/.

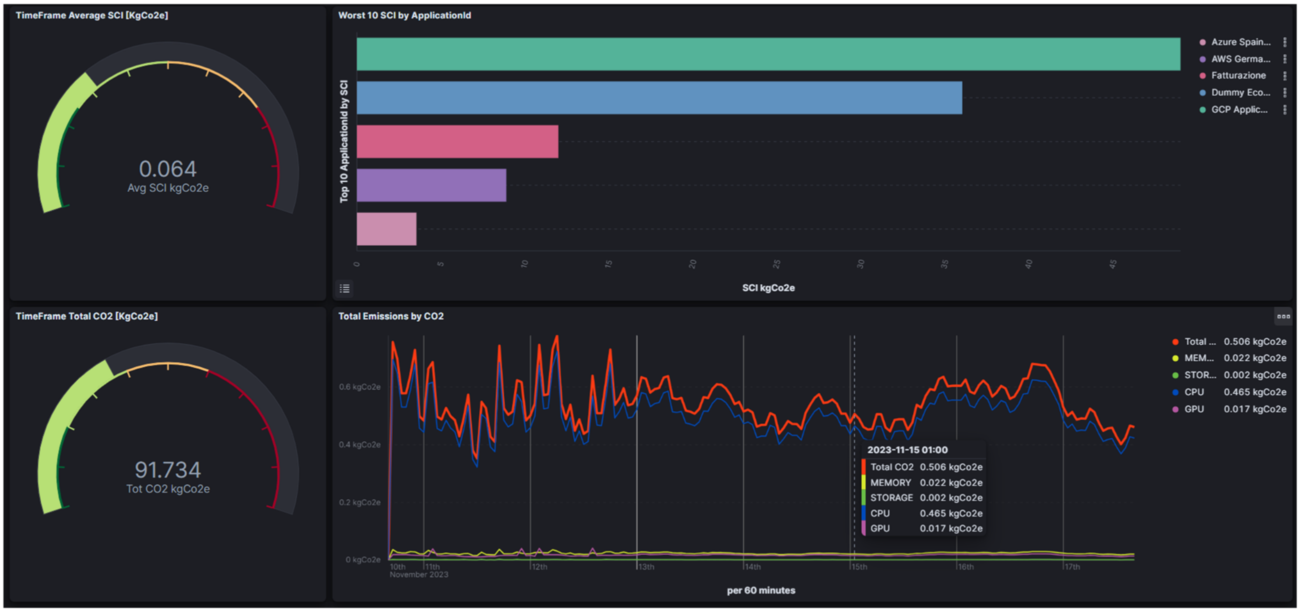


Fig. 11 CO_2eq and SCI monitoring for the bank

applications amounted to 121, 198.46 kgCO_2eq. This corresponds to[4]:

- 134.1 m^3 of trees required for carbon absorption,
- Emissions equivalent to the impact of a car driven for 843, 413.1 km,
- The emissions associated with 207.3 round-trip flights in Europe (HEL-MUC),
- Consumption of 59,927.21 of gasoline.

Consequently, the top 33% of software applications, comprising the first 25 applications with the highest total CO_2eq emissions in the month, were identified. To assess the effectiveness of emission reduction measures facilitated by the Green Index, code optimization operations were conducted on these applications. Comparative monitoring of emissions between the non-optimized and optimized applications occurred simultaneously.

Within the same month and under identical carbon intensity conditions, a discernible reduction in emissions ranging between 3% and 6% was observed compared to the baseline scenario. This translates to an overall reduction of 4%, equating to a total of 4, 582 kgCO_2eq for a single month. Over a year, this reduction amounts to 54, 984 kgCO_2eq, corresponding to:

- 61.2 m^3 of trees required for CO_2 absorption,
- The avoidance of emissions equivalent to driving a car for 669, 603.9 km,
- The prevention of 93.6 round-trip flights in Europe (HEL-MUC),
- The conservation of 27, 187.2 liters of gasoline.

Comparing the CO2eq emissions associated with an IT change (from an optimization point of view) is not as simple as one might think. It is not enough to work on optimization, but it may rather be necessary to apply carbon awareness actions to move the execution to less impactful times of the day or geographical places, and therefore with a lower Carbon Intensity. Even just by moving 33% of the applications, considered as the most impactful, to France, considering an average of 30 gCO_2eq/kWh, it is possible to lower emissions by 58, 175.26 kgCO_2eq, about 48% less than the initial non-optimized situation.

Afterward, the bank mentioned all the actions that are currently underway, as they are part of an ongoing process aimed at achieving sustainable goals across all aspects of the organization's IT:

- Reduced carbon footprint: by tracking and measuring the energy consumption and carbon emissions of its IT systems, the bank can now actively identify areas and ways to reduce its IT carbon footprint; this is leading to a significant reduction in carbon emissions, helping the bank to achieve its environmental targets and contribute to a more sustainable future.
- Cost savings: by optimizing its software and hardware usage for greater sustainability, the bank can also achieve

---

[4] KgCO_2eq conversions are computed with the tool available at: https://www.openco2.net/en/co2-converter.

significant cost savings. This is because reducing energy consumption and carbon emissions often goes hand in hand with reducing overall resource consumption, such as electricity and cooling, which can lead to lower operational costs.

- Improved efficiency: the use of KPIs such as CO_2eq and SCI enables the bank to monitor the efficiency of its software and hardware, letting it continuously identify areas for improvement, and implement changes to increase efficiency; this helps to reduce the bank's impact on the environment, but it also leads to improved overall efficiency and performance of its IT systems.
- Bank's ESG Reputation: the bank is demonstrating its commitment to sustainability and responsible IT practices, such as increasing customer loyalty, stakeholder engagement, and a positive impact on the overall brand reputation.

In addition to reducing carbon emissions, the bank's adoption of ESC framework also contributed to advancements in the issue of standard setting for software sustainability. By choosing to follow the ESC Framework, the bank was able to identify and adopt best practices for Sustainable IT systems and share these practices with others in the industry.

4.2 Optimizing resource efficiency in kubernetes clusters

This section details the utilization of the "Kube-Green" solution by a prominent multi-utility company to rectify resource inefficiencies within their development Kubernetes cluster. The company aimed to achieve both cost savings and environmental impact by automatically powering off idle resources, scaling down clusters, and reducing energy consumption and associated carbon footprint. "Kube-Green" is a sophisticated tool explicitly designed for resource optimization in Kubernetes clusters, providing a comprehensive approach to identify, manage, and dynamically optimize resources.

The tool continuously monitored the Kubernetes cluster, employing intelligent algorithms to identify idle or unused resources during predefined time slots. The implementation offered two key processes for optimization: automatic power off, where idle pods were automatically suspended or powered off to prevent needless energy consumption, and cluster scaling, which dynamically scaled down the entire Kubernetes cluster during periods of reduced demand.

The deployment of "Kube-Green" resulted in significant benefits for the multi-utility company, including a substantial 40% reduction in the monthly bill, from approximately 1200 euro/month to 800 euro/month. Moreover, the automated power-off mechanism and intelligent cluster scaling significantly reduced energy consumption, aligning with the client's sustainability goals and lowering the overall carbon footprint. The implementation not only addressed identified challenges but also optimized resource usage in harmony with natural fluctuations in demand within a development setting.

This successful implementation illustrates how the iterative use of monitoring and optimization, as exemplified by the "Kube-Green" solution, can transform resource management in Kubernetes clusters. The case study showcases how economic efficiency can be harmonized with environmental responsibility, emphasizing the potential of such solutions for sustainable software development practices.

5 Remarks

In response to the imperative need for a formalized and globally recognized approach to mitigate the environmental impact of computing, the standardization and validation of the ESC framework emerge as foundational endeavors. This rigorous process, undertaken collaboratively by diverse stakeholders, follows a meticulously crafted roadmap to establish universally applicable guidelines for sustainable computing practices and to foster global environmental responsibility in the realm of technology.

Commencing with the identification of this pressing need, the standardization journey would begin with the identification of the pressing need for standardized sustainable computing practices. Diverse working groups, comprising industry experts, environmentalists, policymakers, and researchers, are convened to initiate the standardization process. These groups would engage in comprehensive research, drawing from existing best practices and environmental impact assessments related to computing. The draft standards are subjected to rigorous public consultation, allowing for input from a broad spectrum of stakeholders, including businesses, academics, and advocacy groups. Challenges arising from the dynamic technological landscape and global variations in regulatory frameworks will be navigated to produce a standardized ESC framework.

The standardization process would hinge on achieving consensus, a delicate balance considering the diverse interests and perspectives involved. Challenges emerge from the rapidly evolving technological landscape, requiring the standard to be dynamic and adaptable. Achieving consistency on a global scale is also challenging, given the variations in regulatory environments and cultural differences. Once a consensus is reached, the finalized standard is submitted to a recognized standards organization for formal approval. The approved standard, encompassing guidelines and criteria for sustainable computing, is then published and made available to the public. This standardized framework serves as a universal reference point for organizations seeking to adopt and implement environmentally responsible computing practices.

Post-standardization, the critical validation phase unfolds. This intricate roadmap encompasses environmental impact assessments, standardized metric development, and the establishment of certification bodies. These entities rigorously audit organizations, ensuring compliance with ESC framework metrics through on-site inspections, documentation reviews, and continuous monitoring. Periodic reviews and updates further guarantee the framework's relevance amid technological advancements. Collaboration and knowledge-sharing among stakeholders remain integral to maintaining the integrity of this standardized approach. Thus, the ESC framework not only addresses the immediate need for sustainable computing but also lays the groundwork for a collective and enduring commitment to environmental stewardship in the digital era.

Analytical validation becomes a linchpin in substantiating environmental claims and ensuring the credibility of sustainable initiatives. A key step in the validation roadmap is the development of standardized metrics. These metrics, covering aspects like carbon footprint, energy efficiency, and waste management, provide a quantifiable basis for organizations to assess their adherence to the ESC framework. Certification bodies are established to rigorously audit and verify organizations' compliance with standardized metrics. This involves on-site inspections, documentation reviews, and continuous monitoring to ensure ongoing adherence. The roadmap further includes periodic reviews and updates to the ESC framework. This iterative process ensures that the standards remain relevant and effective in the face of evolving technologies and environmental considerations. Collaboration and knowledge-sharing among industry players, researchers, and policymakers play a pivotal role in maintaining the framework's integrity.

Through this comprehensive process of standardization and analytical validation, the ESC framework not only mitigates the environmental impact of computing but also instills confidence, provides a benchmark for improvement, and guides a collective journey toward a sustainable digital future.

6 Conclusion

One of the critical challenges that companies face in adopting sustainable practices to minimize their environmental impact is to track and measure energy consumption and carbon emissions associated with software and hardware usage. In this paper, we have proposed the ESC framework as a holistic approach to IT sustainability that can help companies reach a new way of assessing sustainability, and track, measure, and reduce carbon emissions associated with software and hardware. The ESC framework has the potential to revolutionize the way we conceptualize, design, and implement computing technologies. By pursuing these avenues of research and implementation, we are confident that this framework will be a useful driver for the transformation toward a greener and more sustainable technological future.

Moving forward, our research agenda will focus on several critical fronts. Firstly, we aim to develop innovative solutions to accurately track and measure carbon emissions of most IT solutions, enhancing the framework's practical applicability. Secondly, standardizing the ESC framework will be a key priority, enabling seamless adoption and integration across various domains. To reinforce the credibility of the ESC framework, rigorous analytical validation will be conducted, demonstrating its efficiency and efficacy in sustainable computing. Additionally, further case studies will be undertaken to showcase real-world benefits and encourage its widespread adoption. Lastly, we are committed to advocating the ESC framework to industry stakeholders and researchers in the field of both sustainability and technology, promoting its integration into green computing practices on a bigger scale.

Acknowledgements We gratefully acknowledge the support and contributions from the Green Software Foundation, which have been instrumental in advancing our research on sustainable software development practices.

Author Contributions AP, GB, DCV, and MVT did conceptualization; AP, GB, and DCV provided methodology, visualized the study and done writing—review and editing; GB gave software; AP and DCV validated the study; AP, DCV, and MVT performed formal analysis; AP investigated the study, administrated the project, and did writing—original draft preparation; GB gave resources; GB and DCV done data curation; AP and MVT supervised the study; MVT done funding acquisition. All authors have read and agreed to the submitted version of the manuscript.

Funding This project was self-funded.

Availability of data and materials Not applicable.

Declarations

Conflict of interest The authors report no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Malmodin J, Lundén D (2018) The energy and carbon footprint of the global ICT and E&M sectors 2010–2015. Sustainability 10(9):3027
2. Freitag C, Berners-Lee M, Widdicks K, Knowles B, Blair GS, Friday A (2021) The real climate and transformative impact of ICT: a critique of estimates, trends, and regulations. Patterns 2(9):1–18
3. Elliot S, Binney D (2008) Environmentally sustainable ICT: developing corporate capabilities and an industry-relevant is research agenda. In: PACIS 2008 proceedings 209
4. Baldé C, D'Angelo E, Luda V, Deubzer O, Kuehr R (2022) Global transboundary e-waste flows monitor. GlobalTBM_webversion_june_2_pages. pdf. United Nations Institute for Training and Research (UNITAR)
5. Loeser F (2013) Green it and green is: definition of constructs and overview of current practices
6. Shenoy P (2023) Energy-efficiency versus carbon-efficiency: What's the difference? ACM SIGENERGY Energy Inform Rev 2(4):1–2
7. Katal A, Dahiya S, Choudhury T (2023) Energy efficiency in cloud computing data centers: a survey on software technologies. Clust Comput 26(3):1845–1875
8. Bharany S, Sharma S, Khalaf OI, Abdulsahib GM, Al Humaimeedy AS, Aldhyani TH, Maashi M, Alkahtani H (2022) A systematic survey on energy-efficient techniques in sustainable cloud computing. Sustainability 14(10):6256
9. KPMG (2011) KPMG international survey of corporate responsibility reporting 2011. KPMG, Amsterdam
10. Bansal P (2005) Evolving sustainably: a longitudinal study of corporate sustainable development. Strateg Manag J 26(3):197–218
11. Hart SL (1995) A natural-resource-based view of the firm. Acad Manag Rev 20(4):986–1014
12. Laplante P, Voas J (2023) "Frameworking" carbon-aware computing research. Computer 56(5):105–108
13. Purvis B, Mao Y, Robinson D (2019) Three pillars of sustainability: in search of conceptual origins. Sustain Sci 14:681–695
14. Hindle A (2016) Green software engineering: the curse of methodology. In: 2016 IEEE 23rd international conference on software analysis, evolution, and reengineering (SANER), vol 5. IEEE, pp 46–55
15. Pang C, Hindle A, Adams B, Hassan AE (2015) What do programmers know about software energy consumption? IEEE Softw 33(3):83–89
16. Pinto G, Castor F, Liu YD (2014) Mining questions about software energy consumption. In: Proceedings of the 11th working conference on mining software repositories, pp 22–31
17. Lannelongue L, Aronson H-EG, Bateman A, Birney E, Caplan T, Juckes M, McEntyre J, Morris AD, Reilly G, Inouye M (2023) Greener principles for environmentally sustainable computational science. Nat Comput Sci 3(6):514–521
18. Robillard MP (2016) Sustainable software design. In: Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering, pp 920–923
19. Kern E, Dick M, Naumann S, Hiller T (2015) Impacts of software and its engineering on the carbon footprint of ICT. Environ Impact Assess Rev 52:53–61
20. Bozzelli P, Gu Q, Lago P (2013) A systematic literature review on green software metrics. VU University, Amsterdam
21. McIntosh-Smith S, Wilson T, Crisp J, Ibarra AÁ, Sessions RB (2011) Energy-aware metrics for benchmarking heterogeneous systems. ACM SIGMETRICS Perform Eval Rev 38(4):88–94
22. Kipp A, Jiang T, Fugini M (2011) Green metrics for energy-aware it systems. In: 2011 International conference on complex, intelligent, and software intensive systems. IEEE, pp 241–248
23. Sinha A, Chandrakasan AP (2001) Jouletrack: a web based tool for software energy profiling. In: Proceedings of the 38th annual design automation conference, pp 220–225
24. Schubert S, Kostic D, Zwaenepoel W, Shin KG (2012) Profiling software for energy consumption. In: 2012 IEEE International conference on Green computing and communications. IEEE, pp 515–522
25. Chowdhury SA, Hindle A (2016) Greenoracle: estimating software energy consumption with energy measurement corpora. In: Proceedings of the 13th international conference on mining software repositories, pp 49–60
26. Dodge J, Prewitt T, Combes R, Odmark E, Schwartz R, Strubell E, Luccioni AS, Smith NA, DeCario N, Buchanan W (2022) Measuring the carbon intensity of AI in cloud instances. In: Proceedings of the 2022 ACM conference on fairness, accountability, and transparency, pp 1877–1894
27. Radovanović A, Koningstein R, Schneider I, Chen B, Duarte A, Roy B, Xiao D, Haridasan M, Hung P, Care N et al (2022) Carbon-aware computing for datacenters. IEEE Trans Power Syst 38(2):1270–1280
28. Fawaz A-H, Mohammed AFY, Laku LIY, Alanazi R (2019) PUE or GPUE: a carbon-aware metric for data centers. In: 2019 21st International conference on advanced communication technology (ICACT). IEEE, pp 38–41
29. Bolla R, Bruschi R, Davoli F, Cucchietti F (2010) Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures. IEEE Commun Surv Tutor 13(2):223–244
30. Kilanioti I, Fernández-Montes A, Fernández-Cerero D, Karageorgos A, Mettouris C, Nejkovic V, Albanis N, Bashroush R, Papadopoulos GA (2019) Towards efficient and scalable data-intensive content delivery: state-of-the-art, issues and challenges. In: High-performance modelling and simulation for big data applications: selected results of the COST action IC1406 cHiPSet, pp 88–137
31. Morley J, Widdicks K, Hazas M (2018) Digitalisation, energy and data demand: the impact of internet traffic on overall and peak electricity consumption. Energy Res Soc Sci 38:128–137
32. Ahmad Ibrahim SR, Yahaya J, Sallehudin H (2022) Green software process factors: a qualitative study. Sustainability 14(18):11180
33. Oyedeji S, Penzenstadler B (2020) Experiences from applying the karlskrona manifesto principles for sustainability in software system design. RE4SuSy@ RE
34. Kern E, Hilty LM, Guldner A, Maksimov YV, Filler A, Gröger J, Naumann S (2018) Sustainable software products-towards assessment criteria for resource and energy efficiency. Future Gener Comput Syst 86:199–210
35. Charter M, Tischner U (2017) Sustainable solutions: developing products and services for the future. Routledge, London
36. Epstein MJ, Buhovac AR (2014) Making sustainability work: best practices in managing and measuring corporate social, environmental, and economic impacts. Berrett-Koehler Publishers, Oakland
37. Watson RT, Boudreau M-C, Chen AJ (2010) Information systems and environmentally sustainable development: energy informatics and new directions for the is community. MIS Q 34:23–38
38. Ayewah N, Pugh W, Hovemeyer D, Morgenthaler JD, Penix J (2008) Using static analysis to find bugs. IEEE Softw 25(5):22–29
39. Negri M, Cagno E, Colicchia C, Sarkis J (2021) Integrating sustainability and resilience in the supply chain: a systematic literature review and a research agenda. Bus Strategy Environ 30(7):2858–2886
40. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

41. Chen P, Gao J, Ji Z, Liang H, Peng Y (2022) Do artificial intelligence applications affect carbon emission performance?-Evidence from panel data analysis of Chinese cities. Energies 15(15):5730
42. Kanbach DK, Heiduk L, Blueher G, Schreiter M, Lahmann A (2023) The Genai is out of the bottle: generative artificial intelligence from a business model innovation perspective. Rev Manag Sci. https://doi.org/10.1007/s11846-023-00696-z
43. Wu C-J, Raghavendra R, Gupta U, Acun B, Ardalani N, Maeng K, Chang G, Aga F, Huang J, Bai C et al (2022) Sustainable AI: environmental implications, challenges and opportunities. Proc Mach Learn Syst 4:795–813
44. Mitchell T (1997) Machine learning. McGraw-Hill Education, New York
45. Gupta M, Agrawal P (2022) Compression of deep learning models for text: a survey. ACM Trans Knowl Discov Data (TKDD) 16(4):1–55
46. Saputri TRD, Lee S-W (2021) Integrated framework for incorporating sustainability design in software engineering life-cycle: an empirical study. Inf Softw Technol 129:106407
47. Lannelongue L, Inouye M (2023) Carbon footprint estimation for computational research. Nat Rev Methods Primers 3(1):9
48. Lannelongue L, Grealey J, Inouye M (2021) Green algorithms: quantifying the carbon footprint of computation. Adv Sci 8(12):2100707
49. Gunnarsson J, Kivioja M, Pyrko J (2009) It must be yeti!–Tracking carbon footprints on the web. In: ECEEE conference proceedings, pp 1811–1819
50. Gombiner J (2011) Carbon footprinting the internet. Consilience 5:119–124
51. Kinkar K, Bhosale P, Kasar A, Gutte V (2022) Carbon footprint analysis: need for green cloud computing. In: 2022 International conference on electronics and renewable systems (ICEARS). IEEE, pp 1–6
52. Cao Z, Zhou X, Hu H, Wang Z, Wen Y (2022) Toward a systematic survey for carbon neutral data centers. IEEE Commun Surv Tutor 24(2):895–936
53. Acun B, Lee B, Kazhamiaka F, Sundarrajan A, Maeng K, Chakkaravarthy M, Brooks D, Wu C-J (2023) Carbon dependencies in datacenter design and management. ACM SIGENERGY Energy Inform Rev 3(3):21–26
54. Bouley D (2011) Estimating a data center's electrical carbon footprint. Schneider Electric White Paper Library, pp 14–22
55. Sehgal R, Mehrotra D, Nagpal R, Sharma R (2022) Green software: refactoring approach. J King Saud Univ Comput Inf Sci 34(7):4635–4643
56. Şanlıalp İ, Öztürk MM, Yiğit T (2022) Energy efficiency analysis of code refactoring techniques for green and sustainable software in portable devices. Electronics 11(3):442
57. Guzmán IG-R, Piattini M, Pérez-Castillo R (2015) Green software maintenance. In: Green in software engineering, pp 205–229
58. Pinto G, Soares-Neto F, Castor F (2015) Refactoring for energy efficiency: a reflection on the state of the art. In: 2015 IEEE/ACM 4th international workshop on green and sustainable software. IEEE, pp 29–35
59. Shu W, Cai K, Xiong NN (2021) Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing. Future Gener Comput Syst 124:12–20
60. Radu L-D (2017) Green cloud computing: a literature survey. Symmetry 9(12):295
61. Calero C, Piattini M (2015) Introduction to green in software engineering. In: Green in software engineering, pp 3–27
62. Uddin M, Rahman AA (2012) Energy efficiency and low carbon enabler green it framework for data centers considering green metrics. Renew Sustain Energy Rev 16(6):4078–4094
63. Harrach C, Geiger S, Schrader U (2020) Sustainability empowerment in the workplace: determinants and effects. In: Sustainability management forum— nachhaltigkeits management forum, vol 28. Springer, pp 93–107
64. Kuzma E, Padilha LS, Sehnem S, Julkovski DJ, Roman DJ (2020) The relationship between innovation and sustainability: a meta-analytic study. J Clean Prod 259:120745
65. Wang Y, Sun X, Guo X (2019) Environmental regulation and green productivity growth: empirical evidence on the porter hypothesis from OECD industrial sectors. Energy Policy 132:611–619

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.