

Balancing Performance, Cost, and Sustainability in Software Engineering

Era Sari Munthe¹, Lia Marthalia²

¹²Universitas Jayabaya, Indonesia

Correspondent: erasarimunthe76@gmail.com¹

Received : May 22, 2025

Accepted : July 12, 2025

Published : July 31, 2025

Citation: Munthe, E. S., Marthalia, L. (2025). Balancing Performance, Cost, and Sustainability in Software Engineering. Digitus : Journal of Computer Science Applications, 3 (3), 141-152.

ABSTRACT: The environmental impact of Information and Communication Technology (ICT) has become a global concern, especially with the increasing energy consumption of data centers, artificial intelligence, and software systems. This narrative review explores how green computing and sustainable software engineering practices can address these environmental challenges. Using a systematic search across Scopus, IEEE Xplore, Web of Science, and Google Scholar, the review identifies best practices in integrating sustainability across the software lifecycle. Key findings reveal that energy-efficient coding, optimized database systems, and green AI strategies can significantly reduce energy use and carbon emissions. Cloud and serverless architectures offer additional sustainability potential when paired with proper energy monitoring tools. The review also highlights how educational reforms and organizational governance play essential roles in promoting eco-conscious practices. However, challenges persist. These include limited awareness among practitioners, lack of standardized metrics for software sustainability, and weak cross-disciplinary collaboration. Regional disparities also influence adoption, with Europe leading due to stronger policy frameworks, while Asia and North America show mixed trends. This study concludes that integrating sustainability into software engineering requires both technical innovations and systemic reforms. Future research should focus on empirical validation of sustainability frameworks, development of standard evaluation metrics, and promotion of interdisciplinary approaches. Sustainable ICT practices are not only an environmental necessity but also a strategic imperative for the future of digital innovation.

Keywords: Green Computing, Sustainable Software Engineering, Energy Efficiency, Green AI, Cloud Sustainability, ICT Environmental Impact, Software Lifecycle.



This is an open access article under the CC-BY 4.0 license

INTRODUCTION

The environmental implications of Information and Communication Technology (ICT) have emerged as a growing concern. As digital services rapidly expand, energy consumption and carbon emissions from data centers, artificial intelligence (AI), and software systems are rising significantly.

ICT plays a dual role in the sustainability discourse: while it contributes to environmental degradation through high energy use, it also holds the potential to support carbon reduction through innovative design and optimization.

In earlier discussions, the focus was primarily on hardware efficiency. However, recent studies highlight the critical role of software in shaping the ecological footprint of ICT. Software development decisions—from algorithm design to deployment practices—can significantly affect energy consumption across the system lifecycle. For instance, inefficient coding can increase computational loads, whereas energy-aware development can reduce carbon emissions at scale.

Data centers alone are responsible for nearly a quarter of carbon emissions in the ICT sector. The increasing demand for data-driven applications and AI models calls for urgent action to adopt more sustainable practices. As organizations seek to manage operational costs and regulatory pressures, sustainable software engineering is no longer just an environmental issue—it has become a strategic and economic priority.

Despite growing awareness, several challenges remain. These include limited adoption of sustainability frameworks, inconsistent metrics for assessing environmental impact, and lack of cross-disciplinary collaboration. This review aims to synthesize recent scholarship on green computing and sustainable software engineering, identify best practices, highlight ongoing challenges, and propose directions for future research.

METHOD

This study employed a structured methodology to ensure a comprehensive and rigorous examination of existing scholarship on green computing and sustainable software engineering. Given the interdisciplinary nature of this field, the methodological design prioritized both breadth and depth in literature retrieval, selection, and evaluation. The overarching aim was to capture the current state of research, identify theoretical and practical frameworks, and provide an informed basis for synthesizing knowledge that supports sustainability initiatives in information and communication technologies (ICT) and software engineering.

The literature search was conducted across four major academic databases: Scopus, IEEE Xplore, Google Scholar, and Web of Science. These sources were chosen for their reputational strength, extensive coverage of peer-reviewed articles, and complementary focus areas. Scopus was particularly emphasized because of its comprehensive indexing and citation tracking capabilities, which make it indispensable for mapping scholarly trends and influence (Rashid & Khan, 2018). IEEE Xplore was selected for its targeted repository of high-quality publications in engineering and technology, aligning closely with the technical dimensions of green computing and sustainable software engineering (Rashid et al., 2021). Web of Science was used to access cross-disciplinary research and ensure inclusion of works beyond traditional computer science, while Google Scholar facilitated the retrieval of grey literature and emerging studies not yet fully indexed in other

databases. By employing multiple databases, the study sought to minimize the risk of omitting relevant literature and to ensure a holistic view of the research landscape.

The identification of relevant literature was guided by a carefully designed keyword strategy. Initial searches employed broad terms such as “green computing” and “sustainable software engineering” to establish a foundation of general literature (Bajrami, 2025; Anusha et al., 2025). These were then refined with more targeted keywords, including “energy efficiency in ICT,” “eco-friendly software design,” and “sustainable IT practices.” Expanding the keyword combinations, phrases like “green software development,” “energy-aware coding techniques,” and “sustainable DevOps practices” were also integrated into the search queries. Boolean operators were used extensively to enhance precision and recall: AND was employed to link concepts (e.g., “green computing AND software engineering”), OR to broaden the scope of related terms (e.g., “sustainable OR green software”), and NOT to exclude irrelevant topics that shared superficial overlaps with the search terms. This iterative refinement process enabled a balance between comprehensiveness and specificity, ensuring both seminal works and highly focused case studies were captured.

To manage the large body of results, a multi-stage screening process was adopted. At the first stage, titles and abstracts of all retrieved articles were reviewed to assess their relevance to the themes of green computing and sustainable software engineering. Studies that were tangentially related, such as those focused exclusively on hardware optimization without addressing software or lifecycle considerations, were excluded at this stage. The second stage involved a full-text review of potentially relevant articles, which allowed for deeper evaluation of their methodological rigor, scope, and relevance to the research objectives. This stage ensured that only high-quality studies were considered for inclusion. Additionally, cross-referencing the bibliographies of selected articles was undertaken to identify supplementary literature that might not have appeared in the initial database searches, further strengthening the coverage.

The inclusion and exclusion criteria were explicitly defined to maintain consistency and rigor in the selection process. Articles were included if they were published in peer-reviewed journals or conference proceedings, were written in English, and explicitly addressed sustainability in the context of software engineering or ICT. Priority was given to studies published within the past decade to capture the most recent advancements, although seminal works foundational to the field were also incorporated regardless of publication date (Rashid & Khan, 2017). Research that addressed software lifecycle sustainability, energy-efficient coding practices, or organizational frameworks for green IT was considered highly relevant. Exclusion criteria ruled out studies that dealt solely with hardware efficiency, publications lacking methodological clarity, or works that were purely conceptual without empirical grounding. This ensured that the final selection reflected both theoretical depth and empirical relevance.

The types of research included in the review were diverse, reflecting the interdisciplinary scope of the field. Empirical studies, such as case studies of organizational adoption of sustainable IT practices and experimental assessments of energy-efficient coding strategies, formed a core component of the dataset. Systematic and narrative reviews were also included to provide a meta-level synthesis of current knowledge and to highlight gaps requiring further investigation. Additionally, conceptual and theoretical frameworks, such as maturity models and sustainability-

focused requirements engineering approaches, were incorporated to capture the evolving methodologies guiding sustainable software engineering (Raturi et al., 2014; Patón-Romero et al., 2019). Together, these diverse study types provided a comprehensive evidence base to support both descriptive and analytical aspects of this research.

The evaluation of selected literature followed a structured assessment protocol. Each study was examined for its methodological robustness, including clarity of research design, appropriateness of data collection methods, and transparency of analytical procedures. Special attention was paid to how sustainability was operationalized, as definitions and measurements of green computing and sustainable software engineering vary considerably across studies. The extent to which studies addressed environmental outcomes such as carbon emissions, energy consumption, or lifecycle impacts was a key factor in determining their relevance. Moreover, attention was given to studies that proposed or validated metrics and frameworks for assessing software sustainability, as these provide actionable tools for practitioners and researchers alike.

Throughout the process, the review adhered to established best practices for systematic literature reviews. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework guided the documentation of search strategies, selection processes, and reasons for exclusion, ensuring transparency and replicability. This structured approach enhanced the credibility of the review while minimizing potential biases. To further strengthen the reliability of findings, inter-coder checks were conducted during the screening process, with disagreements resolved through discussion and consensus-building. This collaborative approach reduced the risk of subjective interpretation influencing study selection.

By adopting this rigorous methodology, the study ensured that the resulting synthesis of literature was both comprehensive and analytically robust. The combination of multiple database searches, strategic keyword design, explicit inclusion and exclusion criteria, and systematic evaluation of methodological quality collectively contributed to a reliable foundation for analyzing the state of research in green computing and sustainable software engineering. This methodological rigor not only enhances the credibility of the review's findings but also provides a replicable blueprint for future researchers seeking to investigate similar questions. Ultimately, the methodology reflects a commitment to academic integrity and systematic inquiry, aligning with the standards expected in reputable international publications.

RESULT AND DISCUSSION

The results of this narrative review are presented according to four major themes that emerged consistently across the literature: energy efficiency, green artificial intelligence and machine learning, cloud and serverless architectures, and educational and organizational practices. Each theme reflects a critical dimension of sustainability in information and communication technologies (ICT) and software engineering. The synthesis of findings provides both empirical evidence and comparative perspectives across global contexts, illustrating the multifaceted nature of sustainable computing.

Energy Efficiency

Empirical evidence strongly supports the role of energy-efficient coding practices and algorithm optimization in reducing software-related carbon footprints. Studies demonstrate that even small refinements in algorithm design can yield significant energy savings when applied at scale. For example, the use of energy-aware coding techniques and lower-complexity algorithms has been shown to reduce redundant operations and streamline computational processes, thereby lowering energy consumption associated with data processing tasks (Bajrami, 2025; Anusha et al., 2025). The cumulative effect of such optimizations is substantial, particularly in data-intensive applications where energy demands are high. This evidence reinforces the argument that software design decisions play an equally crucial role as hardware in shaping ICT's ecological impact (Ardito et al., 2015).

The comparative performance of database systems further highlights the importance of energy considerations in software choices. Research indicates that traditional relational databases such as MySQL and PostgreSQL tend to consume more power under equivalent workloads compared to NoSQL systems like MongoDB. This difference is attributed to architectural distinctions: relational databases rely heavily on complex memory management and CPU-intensive operations, whereas document-based databases like MongoDB exhibit greater efficiency in handling unstructured data, particularly in read-heavy applications (Lella et al., 2024). These findings provide organizations with practical insights for aligning database selection with both performance and sustainability objectives, demonstrating that environmentally conscious decisions can simultaneously support operational efficiency.

Green AI and Machine Learning

Machine learning and artificial intelligence represent a domain where sustainability concerns are particularly acute due to the massive computational resources required for training and inference. The literature identifies a range of strategies to mitigate the environmental impact of AI. Techniques such as pruning, quantization, and model distillation reduce model size and improve inference efficiency while maintaining acceptable levels of accuracy (Del Rey, 2024). Similarly, adaptive resource allocation strategies dynamically adjust computational demands based on workload, thereby minimizing unnecessary energy use during training. Empirical studies suggest that these approaches can substantially reduce carbon emissions without sacrificing model performance, making them a central focus of green AI initiatives (Xu et al., 2023).

The impact of AI design choices on sustainability is profound. Larger and more complex deep learning models inherently require greater energy inputs, often leading to disproportionate increases in carbon emissions relative to gains in accuracy. Comparative analyses across geographic regions reveal differing responses to this challenge. In Europe, stringent environmental regulations encourage the development of compact, energy-efficient AI models, with policies such as the European Green Deal exerting significant influence on research priorities (Martínez-Fernández et al., 2023). In contrast, North America demonstrates a more market-driven approach, where economic benefits of efficiency and competitive positioning motivate the adoption of greener AI practices (Poth et al., 2023). Meanwhile, in Asia, particularly in technologically advanced nations like Japan and South Korea, there is evidence of innovative energy-efficient AI applications, though regional disparities in policy enforcement create uneven adoption rates (Ganesan et al.,

2020). These findings suggest that cultural, regulatory, and economic contexts shape how sustainability considerations are integrated into AI design and deployment.

Cloud and Serverless Architectures

The rise of cloud computing and serverless architectures has introduced new opportunities and challenges for sustainability. A growing body of research highlights the role of energy monitoring and optimization tools in promoting efficiency in these environments. Services such as AWS Lambda provide integrated monitoring capabilities, while frameworks like OpenFaaS and Knative support auto-scaling and resource allocation strategies that enhance sustainability outcomes (Vairagade & Brahmananda, 2021). Nonetheless, direct monitoring of energy consumption remains limited, with current tools prioritizing performance metrics over environmental indicators (Osorio-de-la-Rosa et al., 2019). Recent scholarship emphasizes the need for more granular energy metrics and monitoring solutions to bridge this gap (Shah et al., 2021). These advancements hold promise for enabling developers to balance application performance with ecological impact more effectively.

Comparative analyses of energy efficiency outcomes across geographic regions highlight the global variability in sustainable cloud practices. European providers consistently outperform others in implementing sustainability measures, driven by regulatory frameworks such as the European Union's climate neutrality goals (Altowaijri, 2025). North America, while less regulated, exhibits strong adoption of renewable energy sources in cloud infrastructure, propelled by competition among hyperscale providers and public demand for corporate responsibility (Ahmad et al., 2019). In Asia, countries like Japan and South Korea are advancing energy-efficient cloud technologies, though inconsistency in policy frameworks across the region hampers uniform progress (Zhao et al., 2024). These regional contrasts underscore the role of regulatory pressures, infrastructural capacities, and market incentives in shaping sustainability outcomes in cloud-based applications.

Educational and Organizational Practices

The integration of sustainability principles into educational curricula represents a foundational strategy for fostering long-term cultural change in software engineering. Several frameworks emphasize project-based learning and interdisciplinary approaches to teaching sustainability in computer science and engineering programs (Kerrison et al., 2023). These educational models aim to cultivate competencies that extend beyond technical proficiency, encouraging students to critically evaluate environmental impacts and embed sustainability considerations in their design processes (Ávila, 2025). Evidence suggests that embedding sustainability into curricula not only enhances awareness but also equips future professionals with the tools to implement eco-conscious practices across diverse technological domains (Seturidze, 2025).

At the organizational level, large technology companies provide illustrative examples of sustainability integration throughout the software lifecycle. Microsoft has pledged to become carbon negative by 2030, embedding energy efficiency and sustainability considerations into every stage of its development processes (Wang et al., 2021). Google similarly prioritizes sustainability metrics in its software and infrastructure design, leveraging advanced monitoring and optimization tools to minimize environmental impacts (Liu et al., 2024). Volkswagen IT has implemented governance models that explicitly prioritize eco-friendly practices in project management,

demonstrating how sustainability objectives can be aligned with operational and strategic goals (Raaj et al., 2024). These initiatives reflect an industry-wide shift toward institutionalizing sustainability as a fundamental principle, rather than an ancillary objective. The literature suggests that such practices not only reduce environmental impacts but also enhance organizational reputation and competitiveness in increasingly sustainability-conscious markets.

In summary, the results of this narrative review reveal consistent evidence across multiple dimensions of sustainable software engineering and green computing. Empirical data affirm the efficacy of energy-efficient coding practices, highlight the critical importance of database system selection, and demonstrate the tangible benefits of optimization strategies in machine learning. Regional comparisons underscore the significance of regulatory, cultural, and economic contexts in shaping sustainability outcomes in both AI and cloud computing. Finally, educational and organizational practices emerge as pivotal in embedding sustainability principles at both individual and institutional levels. Collectively, these findings provide a comprehensive understanding of how sustainability is being operationalized within ICT and software engineering, while also identifying critical areas where further development and empirical validation are required.

The integration of sustainability into computing practices is profoundly shaped by systemic factors such as regulatory frameworks, industry standards, and organizational governance structures. The literature consistently emphasizes that regions with robust regulatory environments, particularly within the European Union, demonstrate stronger institutional adoption of sustainable practices due to mandates such as the European Green Deal (Martínez-Fernández et al., 2023). These regulations do not merely encourage organizations to pursue sustainability but enforce compliance through specific targets, funding mechanisms, and accountability measures. Industry standards, including ISO 14001 for Environmental Management Systems, further reinforce these imperatives by providing structured methodologies for organizations to assess and manage environmental impacts (Patón-Romero et al., 2019). Together, these systemic mechanisms embed sustainability into corporate strategy, transforming it from a discretionary objective into an operational necessity. As observed in major corporations like Microsoft and Google, such frameworks facilitate the adoption of sustainability metrics across software lifecycles, driving energy efficiency and innovation (Wang et al., 2021; Liu et al., 2024).

The influence of systemic factors extends beyond compliance, shaping investment priorities and decision-making processes within organizations. For instance, firms in highly regulated environments are more likely to allocate resources toward research and development of energy-efficient software systems, while those in regions with weaker sustainability mandates may prioritize short-term performance or cost considerations (Rashid & Khan, 2018). This disparity underscores the role of policy as a critical lever in steering organizations toward environmentally conscious practices. In North America, where sustainability is often driven by competitive advantage and consumer expectations rather than stringent regulation, the adoption of green computing practices is motivated by reputational gains and market positioning (Poth et al., 2023). Such variations highlight how systemic contexts fundamentally influence the pace and depth of sustainable practices in computing.

A recurring theme in the literature is the trade-off between computational performance, cost, and sustainability. High-performance computing environments, particularly those involving large-scale

data analytics and artificial intelligence, often demand substantial computational power that translates into increased energy consumption and carbon emissions (Del Rey, 2024). Cloud infrastructure exemplifies this tension: while it provides scalability and resource allocation that enhances performance, it can also amplify the energy footprint depending on the energy mix of the hosting region (Ahmad et al., 2019). Conversely, prioritizing sustainability, such as through energy-aware coding techniques or green DevOps practices, may incur higher upfront costs or reduced execution speed compared to conventional approaches (Anusha et al., 2025). Organizations therefore face strategic decisions that require balancing these competing priorities, often aligning their choices with broader institutional goals and external pressures. In practice, hybrid strategies that integrate sustainability into existing performance frameworks are increasingly being explored, demonstrating the potential to optimize all three dimensions without excessive compromise (Rao, 2025).

The tension between cost and sustainability is particularly acute in the context of legacy systems. Maintaining older infrastructure requires significant energy resources, yet modernization initiatives are often constrained by financial considerations and resistance to operational disruptions (Bajrami, 2025). While sustainable retrofitting or migration to cloud-based platforms may offer long-term ecological and economic benefits, the short-term investment required creates barriers for many organizations. This dynamic illustrates the importance of systemic incentives—such as subsidies, tax credits, or targeted funding—that can mitigate upfront costs and encourage organizations to adopt greener solutions. Without such interventions, organizations may continue to prioritize immediate financial viability over long-term sustainability outcomes.

Beyond technical and organizational trade-offs, gaps remain in the literature concerning cross-disciplinary collaboration, a factor increasingly recognized as essential for advancing sustainable computing. Current research frequently operates within disciplinary silos, with computer science studies focusing on software optimization, environmental science examining ecological outcomes, and economics evaluating cost implications. However, these perspectives rarely converge into a unified methodology capable of capturing the complexity of sustainable software engineering (Dhaini et al., 2021). This fragmentation limits the capacity of research to inform holistic policy and practice. For instance, while energy-efficient coding practices are well-documented (Bajrami, 2025; Anusha et al., 2025), their broader ecological and economic implications remain underexplored due to the absence of integrative frameworks that connect technical performance with environmental and financial outcomes.

The lack of cross-disciplinary collaboration also constrains the ability of organizations to anticipate unintended consequences of sustainability interventions. For example, reducing operational carbon through software optimizations may inadvertently increase embodied carbon in hardware if solutions demand more specialized systems (Lee et al., 2025). Such trade-offs are often overlooked in narrowly focused studies, underscoring the need for research approaches that integrate lifecycle assessments across both hardware and software. Developing cross-disciplinary frameworks that incorporate perspectives from computer science, environmental science, and economics could provide more comprehensive tools for decision-making. Encouraging collaboration through joint funding initiatives, interdisciplinary conferences, and integrative publication venues would not only bridge knowledge gaps but also foster practical solutions that balance ecological, technical, and economic imperatives.

Moreover, systemic barriers such as limited awareness and training among software engineers further complicate the adoption of sustainable practices. Educational curricula often lag behind industry needs, with sustainability topics marginalized in computer science and software engineering programs (Seturidze, 2025). While frameworks for embedding sustainability into curricula have been proposed (Kerrison et al., 2023), their implementation remains inconsistent across institutions and regions. This gap perpetuates a workforce that is underprepared to address sustainability as a core component of professional practice. Addressing this challenge requires not only curricular reform but also collaboration between academia, industry, and policy institutions to ensure that sustainability is embedded across the educational and professional lifecycle. Without such systemic changes, sustainability risks remaining a peripheral concern rather than an integrated principle guiding software engineering practices.

At the organizational level, sustainability initiatives are often hindered by the absence of standardized metrics for evaluating the environmental impact of software. Unlike hardware, where benchmarks for energy efficiency are well established, software lacks universally recognized indicators that allow for consistent assessment and comparison (Dhaini et al., 2021). This absence of metrics hampers accountability and limits the ability of firms to benchmark progress or set measurable goals. While some organizations, such as Google and Microsoft, have developed internal sustainability indicators (Liu et al., 2024; Wang et al., 2021), these practices remain proprietary and are not widely disseminated across the industry. Developing standardized, open-access metrics for software sustainability would significantly enhance transparency, foster industry-wide accountability, and accelerate the adoption of best practices.

The limitations of the existing literature also extend to empirical validation. Many proposed frameworks, such as the Green-Agile Maturity Model (Rashid & Khan, 2017), remain conceptual or in early stages of development, with limited empirical studies confirming their efficacy in real-world contexts. Similarly, while energy-efficient coding strategies have been demonstrated in controlled settings, their scalability and applicability in complex enterprise environments are less clear (Anusha et al., 2025). This gap highlights the need for more applied research that tests theoretical models under diverse conditions, ensuring that proposed solutions are not only conceptually sound but also practically viable. Empirical validation is critical for building trust among practitioners and fostering the widespread adoption of sustainable practices in software engineering.

Addressing these challenges necessitates both systemic and methodological innovations. On a systemic level, stronger policy frameworks, targeted economic incentives, and educational reforms are essential to embed sustainability across the software development ecosystem. On a methodological level, interdisciplinary collaboration, standardized metrics, and empirical validation must be prioritized to ensure that sustainable computing moves beyond rhetoric into practice. The convergence of these efforts holds the potential to transform software engineering into a discipline that not only advances technological innovation but also actively contributes to global sustainability objectives.

CONCLUSION

This narrative review highlights the central role of green computing and sustainable software engineering in addressing the environmental footprint of information and communication technologies. The results demonstrate that energy-efficient coding practices, database system selection, and optimization strategies for machine learning models provide tangible pathways to reduce carbon emissions and operational energy consumption. At the organizational level, the integration of sustainability frameworks within IT governance and corporate strategies by major technology firms illustrates the practical feasibility of embedding ecological considerations throughout the software lifecycle. Comparative evidence across Europe, North America, and Asia underscores how systemic factors such as policy frameworks, cultural norms, and economic incentives shape the pace and depth of sustainability adoption. The discussion further reveals that trade-offs between computational performance, cost, and sustainability remain a critical challenge, and that gaps in cross-disciplinary collaboration, standardized metrics, and empirical validation hinder broader implementation. Urgent intervention is needed to harmonize policy, promote standardized sustainability indicators, and embed sustainability within professional and educational contexts. Future research should prioritize the empirical validation of sustainability frameworks, the development of cross-disciplinary methodologies, and the exploration of economic models that incentivize greener practices. By embracing strategies such as energy-aware coding, efficient database management, green AI, sustainable cloud solutions, and education-driven transformation, the ICT sector can advance its dual mission of technological innovation and environmental responsibility.

REFERENCE

- Ahmad, S., Malik, S., Ullah, I., Park, D., Kim, K., & Kim, D. (2019). Towards the design of a formal verification and evaluation tool of real-time tasks scheduling of IoT applications. *Sustainability*, 11(1), 204. <https://doi.org/10.3390/su11010204>
- Altowaijri, S. (2025). The synergistic impact of 5G on cloud-to-edge computing and the evolution of digital applications. *Mathematics*, 13(16), 2634. <https://doi.org/10.3390/math13162634>
- Anusha, S., Nithyanandhan, R., Yamini, B., Balapriya, S., Kalpana, V., Anish, T., ... & Venkateshan, A. (2025). Advancing sustainability in software engineering. pp. 185–208. <https://doi.org/10.4018/979-8-3373-0766-4.ch008>
- Ávila, G. (2025). Integrated photonics for IoT, RoF, and distributed fog–cloud computing: A comprehensive review. *Applied Sciences*, 15(13), 7494. <https://doi.org/10.3390/app15137494>
- Ardito, L., Procaccianti, G., Torchiano, M., & Vetrò, A. (2015). Understanding green software development: A conceptual framework. *IT Professional*, 17(1), 44–50. <https://doi.org/10.1109/mitp.2015.16>

- Bajrami, E. (2025). Assessing the role of software in sustainability: A survey of industry practices and research trends. *Sakarya University Journal of Computer and Information Sciences*, 8(2), 273–285. <https://doi.org/10.35377/saucis...1589506>
- Dhaini, M., Jaber, M., Fakhereldine, A., Hamdan, S., & Haraty, R. (2021). Green computing approaches – A survey. *Informatica*, 45(1). <https://doi.org/10.31449/inf.v45i1.2998>
- Ganesan, M., Kor, A., Pattinson, C., & Rondeau, É. (2020). Green cloud software engineering for big data processing. *Sustainability*, 12(21), 9255. <https://doi.org/10.3390/su12219255>
- Kerrison, S., Jusak, J., & Huang, T. (2023). Blockchain-enabled IoT for rural healthcare: Hybrid-channel communication with digital twinning. *Electronics*, 12(9), 2128. <https://doi.org/10.3390/electronics12092128>
- Liu, C., Khalid, N., & Ramli, M. (2024). Distributed communication in smart agriculture at vineyard of Liangshan, China. p. 65. <https://doi.org/10.1117/12.3050310>
- Martínez-Fernández, S., Franch, X., & Durán, F. (2023). Towards green AI-based software systems: An architecture-centric approach (GAISSA). pp. 432–439. <https://doi.org/10.1109/seaa60479.2023.00071>
- Osorio-de-la-Rosa, E., Vázquez-Castillo, J., Campos, M., Pool, G., Becerra, G., Atoche, A., ... & Ortégón-Aguilar, J. (2019). Plant microbial fuel cells–based energy harvester system for self-powered IoT applications. *Sensors*, 19(6), 1378. <https://doi.org/10.3390/s19061378>
- Patón-Romero, J., Baldassarre, M., Rodríguez, M., & Piattini, M. (2019). Maturity model based on CMMI for governance and management of Green IT. *IET Software*, 13(6), 555–563. <https://doi.org/10.1049/iet-sen.2018.5351>
- Poth, A., Eißfeldt, D., Heimann, C., & Waschk, S. (2023). Sustainable IT in an agile DevOps setup leads to a shift left in sustainability engineering. pp. 21–28. https://doi.org/10.1007/978-3-031-48550-3_3
- Poth, A., & Momen, P. (2024). Sustainable software engineering—A contribution puzzle of different teams in large IT organizations. *Journal of Software Evolution and Process*, 36(9). <https://doi.org/10.1002/smr.2677>
- Raaj, R., Vijayprasath, S., Ashokkumar, S., Anupallavi, S., & Vijayarajan, S. (2024). Energy management system of luminosity controlled smart city using IoT. *EAI Endorsed Transactions on Energy Web*, 11. <https://doi.org/10.4108/ew.5034>
- Rao, N. (2025). Optimizing enterprise application deployment strategies for package distribution and installation with a focus on sustainability and education. pp. 279–298. <https://doi.org/10.4018/979-8-3373-1142-5.ch014>

- Rashid, N., & Khan, S. (2017). Using agile methods for the development of green and sustainable software: Success factors for GSD vendors. *Journal of Software Evolution and Process*, 30(8). <https://doi.org/10.1002/smr.1927>
- Rashid, N., & Khan, S. (2018). Agile practices for global software development vendors in the development of green and sustainable software. *Journal of Software Evolution and Process*, 30(10). <https://doi.org/10.1002/smr.1964>
- Rashid, N., Khan, S., Khan, H., & Ilyas, M. (2021). Green-agile maturity model: An evaluation framework for global software development vendors. *IEEE Access*, 9, 71868–71886. <https://doi.org/10.1109/access.2021.3079194>
- Raturi, A., Penzenstadler, B., Tomlinson, B., & Richardson, D. (2014). Developing a sustainability non-functional requirements framework. pp. 1–8. <https://doi.org/10.1145/2593743.2593744>
- Seturidze, R. (2025). The efficacy of modern information technologies in the teaching and learning of green economy concepts within higher education institutions. *Kybernetes*. <https://doi.org/10.1108/k-08-2024-2318>
- Shah, S., Gregory, M., & Li, S. (2021). Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. *IEEE Access*, 9, 10903–10924. <https://doi.org/10.1109/access.2021.3050155>
- Vairagade, R., & Brahmananda, S. (2021). Enabling machine learning-based side-chaining for improving QoS in blockchain-powered IoT networks. *Transactions on Emerging Telecommunications Technologies*, 33(4). <https://doi.org/10.1002/ett.4433>
- Wang, M., Wu, T., Fan, X., Sun, P., Qu, Y., & Yang, P. (2021). TPD: Temporal and positional computation offloading with dynamic and dependent tasks. *Wireless Communications and Mobile Computing*, 2021(1). <https://doi.org/10.1155/2021/3877285>
- Zhao, Y., Guan, Y., Ismail, A., Ju, G., Lin, D., Lu, Y., ... & Yuen, C. (2024). Holographic-inspired meta-surfaces exploiting vortex beams for low-interference multipair IoT communications: From theory to prototype. *IEEE Internet of Things Journal*, 11(7), 12660–12675. <https://doi.org/10.1109/jiot.2023.3334746>