


Chapter 5

Carbon Footprint

Analysis of Software

Systems

P. Selvakumar

 <https://orcid.org/0000-0002-3650-4548>


*Department of Science and Humanities, Nehru Institute of Technology,
Coimbatore, India*

T. C. Manjunath

 <https://orcid.org/0000-0003-2545-9160>

Rajarajeswari College of Engineering, India

Abhijeet Das

 <https://orcid.org/0000-0003-4599-5462>

C.V. Raman Global University, Bhubaneswar, India

ABSTRACT

In recent years, the term “carbon footprint” has emerged as a critical metric in understanding and addressing the environmental impact of human activities. Broadly defined, a carbon footprint refers to the total amount of greenhouse gases (GHGs), primarily carbon dioxide (CO₂), emitted directly or indirectly by an individual, organization, product, or activity, expressed in equivalent tons of CO₂. This measure includes emissions produced by burning fossil fuels for energy, transportation, manufacturing, and other industrial processes. The concept has gained increasing prominence as the global community intensifies efforts to combat climate change and reduce the accumulation of greenhouse gases that contribute to global warming and its associated adverse effects such as rising sea levels, extreme weather

DOI: 10.4018/979-8-3373-4652-6.ch005

Copyright © 2026, IGI Global Scientific Publishing. Copying or distributing in print or electronic forms without written permission of IGI Global Scientific Publishing is prohibited. Use of this chapter to train generative artificial intelligence (AI) technologies is expressly prohibited. The publisher reserves all rights to license its use for generative AI training and machine learning model development.

events, and biodiversity loss. Traditionally, discussions on carbon footprints have centered around sectors with visible and tangible environmental impacts—such as transportation, agriculture, manufacturing, and energy production.

INTRODUCTION TO CARBON FOOTPRINT

In recent years, the term “carbon footprint” has emerged as a critical metric in understanding and addressing the environmental impact of human activities. Broadly defined, a carbon footprint refers to the total amount of greenhouse gases (GHGs), primarily carbon dioxide (CO₂), emitted directly or indirectly by an individual, organization, product, or activity, expressed in equivalent tons of CO₂. This measure includes emissions produced by burning fossil fuels for energy, transportation, manufacturing, and other industrial processes. The concept has gained increasing prominence as the global community intensifies efforts to combat climate change and reduce the accumulation of greenhouse gases that contribute to global warming and its associated adverse effects such as rising sea levels, extreme weather events, and biodiversity loss. Traditionally, discussions on carbon footprints have centered around sectors with visible and tangible environmental impacts—such as transportation, agriculture, manufacturing, and energy production. However, with the rapid advancement and integration of digital technologies in every facet of life, a growing concern has surfaced regarding the environmental footprint of software systems and digital infrastructures. While software itself may appear intangible and invisible compared to physical industries, the underlying hardware, data centers, and energy consumption associated with running software applications have a significant carbon footprint. This growing realization has prompted researchers, developers, and policymakers to investigate the environmental consequences of software development, deployment, and usage, highlighting the need for sustainable practices within the software industry. Software systems, encompassing everything from mobile apps and websites to large-scale cloud computing platforms and artificial intelligence models, rely heavily on data centers and network infrastructure that consume vast amounts of electricity.

Moreover, the software development lifecycle itself can contribute to carbon emissions through the consumption of computational resources during coding, testing, debugging, and deployment processes. For example, automated testing frameworks, continuous integration/continuous deployment (CI/CD) pipelines, and extensive use of cloud-based development tools require substantial compute power. The rise of computationally intensive applications such as machine learning, big data analytics, and blockchain further exacerbates this trend, as these technologies demand significant processing power and storage capacity. These activities cumulatively increase

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/carbon-footprint-analysis-of-software-systems/394145

Related Content

Validation and Verification of Numerical Models

Bahaa Hayder Mohammed, Yas Khudhair Abbas, Basava Ramanjaneyulu Gudivakaand Sri Harsha Grandhi (2024). *Coding Dimensions and the Power of Finite Element, Volume, and Difference Methods* (pp. 248-273).

www.irma-international.org/chapter/validation-and-verification-of-numerical-models/352316

Towards Automation of IoT Analytics: An Ontology-Driven Approach

Sounak Deyand Arijit Mukherjee (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications* (pp. 947-971).

www.irma-international.org/chapter/towards-automation-of-iot-analytics/188242

Expansion and Practical Implementation of the MFC Cybersecurity Model via a Novel Security Requirements Taxonomy

Neila Rjaibiand Latifa Ben Arfa Rabai (2015). *International Journal of Secure Software Engineering* (pp. 32-51).

www.irma-international.org/article/expansion-and-practical-implementation-of-the-mfc-cybersecurity-model-via-a-novel-security-requirements-taxonomy/142039

Assistive Technology-Based Solution for Hearing Impairment Using Smartphones

Vasu Mehraand Dhiraj Pandey (2022). *International Journal of Software Innovation* (pp. 1-17).

www.irma-international.org/article/assistive-technology-based-solution-for-hearing-impairment-using-smartphones/292024

Security Requirements Engineering for Evolving Software Systems: A Survey

Armstrong Nhlabatsi, Bashar Nuseibehand Yijun Yu (2010). *International Journal of Secure Software Engineering* (pp. 54-73).

www.irma-international.org/article/security-requirements-engineering-evolving-software/39009