

Hybrid Green CI/CD Framework: Real-Time Carbon-Aware Scheduling for Sustainable Software Delivery

Samarth B Kalyani
Department of CSE
BNM Institute of Technology
Bengaluru, India
Email: samarthk007x@gmail

Sharvari H M
Department of CSE
BNM Institute of Technology
Bengaluru, India
Email: hmsharvari@gmail.com

Harshit C
Department of CSE
BNM Institute of Technology
Bengaluru, India
Email: harshitchandrakanth@gmail.com

Dr.Swetha M D Department
of CSE
BNM Institute of Technology
Bengaluru, India
Email: swetha.md@bnmit.in

Abstract—In this paper, an intelligent, automated framework is presented for the integration of real time carbon intensity data into CI/CD pipelines, thereby making software development practices environmentally friendly. In this regard, the authors propose a Hybrid Green CI/CD Framework that utilizes APIs, real time environmental metrics, custom scheduling algorithms, and policy mechanisms to ensure that software automation is transformed into a green, responsible process. Experimental results using GitHub Actions are presented, showing the potential for minimizing the environmental footprint of traditional software development practices.

Index Terms—Green CI/CD, Carbon Aware Scheduling, Carbon Intensity, DevOps Automation, Energy Efficient Computing, Software Carbon Intensity (SCI)

I. INTRODUCTION

Environmental sustainability is one of the prominent outcomes of the increasing rate of cloud native applications and digital infrastructure deployment in the field of software engineering. In recent times, with the increasing reliance of contemporary software systems on Continuous Integration and Continuous Deployment (CI/CD) pipelines, the significant influence of CI/CD pipelines on energy consumption and carbon emissions is becoming more and more widely recognized. Conventional CI/CD pipelines, which are usually linear pipelines of build, test, and deployment jobs, are not designed to take into account the dynamic and time varying carbon intensity of the energy grid. This can cause energy intensive jobs to occur when the grid is highly dependent on fossil fuels, thereby causing unnecessary emissions and making it difficult to achieve net zero emissions goals.

Green Software Engineering is a new field that was created to help mitigate some of these issues by integrating objective and data driven environmentally based metrics into DevOps. It looks at the delivery of software as a challenge to optimize it

from a technical standpoint as well as take advantage of the delivery of software as a form of real time stewardship of the environment. Carbon aware scheduling is an important part of this vision, as it involves scheduling the use of computing resources based on real time emissions data from the electricity grid. This has significant implications for CI/CD workflows that do not currently consider sustainability as a priority when delivering new features and functionality rapidly and reliably.

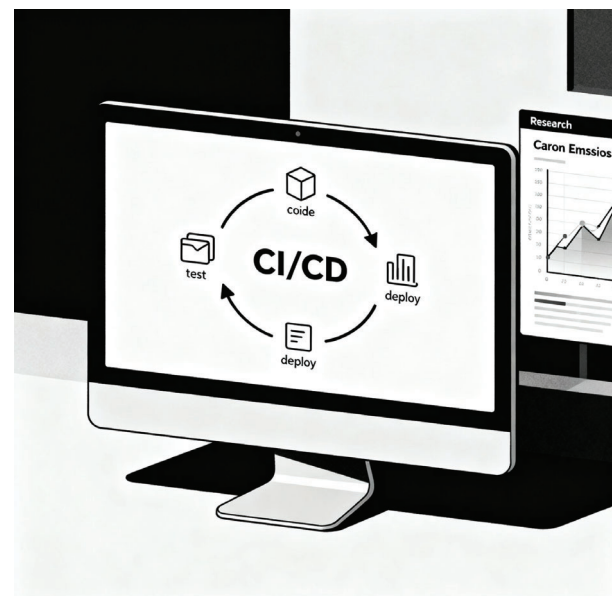


Fig. 1. CI/CD with carbon emissions monitoring.

Hybrid Green CI/CD Framework is a framework that cap-

tures the shift in the above requirements through the introduction of a dynamic and intelligent system for the management of real time emissions. Unlike the conventional approach of analyzing the pipeline and taking action retrospectively, the framework allows for the enforcement of policies through the execution or deferral of jobs based on real time carbon intensity data retrieved from public APIs. This makes CI/CD pipelines not just mere executors but rather agents of environmental policy.

With the integration of sustainability checks within the DevOps automation process, the proposed framework improves operational efficiency and sustainability. This paper has shown the feasibility and need for the integration of carbon awareness within daily software development processes, leading the way for carbon neutral digital innovations in the industry.

II. LITERATURE REVIEW

In fact, the current methods of software development, especially those based on CI/CD, have revolutionized the speed, reliability, and scalability of digital solutions. However, it is now recognized that these continuously running CI/CD pipelines triggered on every commit of the code, often running on resource intensive cloud infrastructure, are significant contributors to global energy demands and carbon emissions. In fact, with digital transformation happening in all sectors of the world, the importance of the environmental effects of continuously running software builds, tests, and deployments has been increasingly recognized by researchers.

The initial research in green software engineering aimed at quantifying the power consumption of IT infrastructure or coding practices. Teams started monitoring cloud data centers and coding tools for their power consumption profiles, which led to discussions on sustainability metrics in software delivery. Sustainable DevOps takes this initial research a step further by aiming at embedding energy aware logic in all aspects of software delivery, including infrastructure or coding practices.

The advent of carbon aware CI/CD pipelines marks a change in this domain. Researchers have proposed and prototyped frameworks where real time carbon intensity data from national grids or third party APIs can be integrated into DevOps pipelines. To illustrate this, Claßen et al. (2023) demonstrated the technical and operational viability of implementing deferral strategies for build and test operations when the carbon intensity is high through the use of a plug in model for GitHub Actions based on UK Government carbon data. Their large scale study found that even partial implementation (such as skipping routine operations during peak hours) resulted in measurable carbon savings without impacting the performance of the operations.

Analogous technologies have also been developed in the domain of cloud native computing. In the paper by Pinna-pareddy et al. (2025), the authors developed a scheduling system for Kubernetes clusters that can be used to schedule the geographically and temporally shifting of heavy resource workloads based on the availability of green energy using

ElectricityMap and WattTime APIs. The authors' experiments with the scheduling system in enterprise settings show that strategically scheduling workloads can help reduce operational CO2 emissions by as much as 30%, providing cloud providers and enterprises with an opportunity to go green with the delivery of software services.

Data driven algorithm selection is also becoming a new hot area of research. Instead of using a one size fits all scheduling strategy, researchers are now suggesting the use of adaptive pipelines that can select the most optimal deferral strategy using historical and real time data. Bostandoost et al. (2025) proved that using an adaptive system with multiple algorithms is always better than using a single algorithm in minimizing the carbon footprint. By using algorithm selection in the CI/CD pipeline, we can dynamically respond to changing grid conditions and release patterns.

Such innovations are associated with energy efficient coding and development methods as well. Saleem et al(2023) and others have shown that fine tuning of application logic, reduction of computational redundancies, and modularization of the code can make each build of the application less energy intensive. Thus, the cumulative benefits of efficient coding and intelligent CI/CD can be leveraged when automation is ubiquitous in large organizations.

These are the steps identified by suitable expert analysts as required for making carbon aware pipelines ubiquitous and a viable option for customers. This includes creating dashboards to transparently track the pipelines' emissions and enable teams to achieve their sustainability goals in a fun way. Changes to policy processes such as running builds only on relevant changes, leveraging green runner infrastructure, and reducing wasteful resources through transient environments, also contribute greatly.

The combination of environmental concerns, modeling tools to forecast future events, and automated cloud processing is creating new benchmarks for industries. Case studies from major technology vendors in addition to CGI UK have demonstrated that businesses can utilize secure and reliable access to live carbon measurements in their critical delivery systems. Early examples of this can be seen within several government cloud platforms where organisations are proving how carbon smart DevOps can be used as an operational tool to help achieve net zero.

However, literature still identifies some challenges in the field. These are the accuracy and detail of the carbon intensity data, its consistency across different regions, and how well sustainability tools integrate with different CI/CD tools. There is also the cultural shift needed in software engineering teams. While standards like the Green Software Foundation's Software Carbon Intensity (SCI) are gaining traction, scientists are advocating for global collaborations to improve auditing tools and encourage green behaviors.

To sum up, green CI/CD and sustainable DevOps have many different types of research that will continue to grow. Technical solutions include everything from real time scheduling through adaptive algorithms and hardware aware coding, which are in-

creasingly supported by experiments in real world enterprises. The studies reviewed appear to demonstrate that the use of carbon aware automation will be common in the near term and considered fundamental to today's software delivery.

III. CHALLENGES

The implementation of a Green CI/CD framework comes with several significant challenges that need to be addressed to ensure effective, sustainable, and scalable software delivery.

A. Integration of Real Time Environmental Data

Integrating live carbon intensity data into CI/CD pipelines requires you to seamlessly integrate environment APIs with your build pipelines. The key challenge in this case is how you handle asynchronous data requests without slowing down or blocking your pipeline because a failure in pulling data can affect your entire build/deployment pipeline.

B. Complexity of Conditional Scheduling Logic

To create intelligent scheduling algorithms which can both automatically postpone/approve build activities according to their dynamic carbon threshold level will require clear policy definitions. The sustainable balance of these algorithms shall not only avoid the cost and time associated with waiting on an approved request, but will also achieve the greatest amount of benefit to the environment.

C. Variability and Accuracy of Carbon Intensity Data

The variable nature of the carbon intensity of the grid, as well as regional differences and the availability of data, makes the decision making process in the scheduling stage complex. One of the challenges in the decision making process in the scheduling stage is the accuracy and availability of the data.

D. Scalability and Portability of the Solution

To create a framework that can scale from individual projects to enterprise environments without compromising performance, it is important to have a strong standardization and architectural design that is robust enough to be easily incorporated into multiple CI/CD platforms.

E. User Adoption and Operational Complexity

However, introducing carbon aware scheduling into current DevOps pipelines requires a change in the current practices of both development and operations teams. Resistance to change, a new toolset to learn, and the potential increase in pipeline complexity are all potential barriers to adoption.

F. Accurate Emissions Estimation and Verification

Measuring the actual carbon footprint of each build job means measuring not just the power consumed by your hardware, but also how long it took for that job to execute. It is hard to create credible and accurate sustainability metrics that stakeholders can trust to be transparent and reportable.

G. Data Privacy and Security Concerns

The inclusion of realtime carbon intensity information or environmental API services within CI/CD pipelines can pose risks in terms of privacy and security. In this case, critical operational information can be made available to third party services, while the number of API calls required for a realtime update can add more complexity in terms of security risks for the system. Therefore, a new dimension of complexity must be addressed in terms of privacy and security while ensuring seamless operation within CI/CD pipelines for a green DevOps approach.

The multidisciplinary and technical challenges inherent in the transition from traditional CI/CD pipelines towards environmentally responsible software engineering are evidenced by these challenges. Thus, addressing these issues enables green DevOps practices to be implemented with greater efficacy in the real world.

IV. EXISTING RESEARCH TRENDS

Integration of sustainability in the domain of software development pipelines like Continuous Integration and Continuous Deployment has picked up momentum as a key area in green software engineering. A wide range of recent research has been conducted on the domain of carbon aware computing and its ability to reduce the footprint of software development pipelines without compromising performance and speed.

Claßen and colleagues (2023) have created an integrated carbon aware CI/CD framework that uses real time and historical carbon emissions data to calculate the most optimal times to schedule builds and tests. Claßen showed at the ICSOC workshops that deferring non urgent workflows until times of low grid carbon emissions resulted in significant reductions in carbon emissions while preserving throughput in the pipeline. The findings of their study indicate that combining user provided deadlines with carbon information leads to optimal scheduling of jobs as well as the significance of being aware of deadlines with respect to achieving a balance between sustainability and business requirements[2].

Also contributing to the cloud orchestrator growth, research into cloud native orchestration products such as Kubernetes continues to progress rapidly. The study by Pinnapareddy et al(2025). introduced algorithms to enable carbon aware scheduling of workloads within the Kubernetes environment, thereby allowing the optimization of the workload based upon the carbon emitted for power supplied to the workload to be performed. In their analysis of the subject area, they indicated there were almost an 80% reduction in both energy and emissions by optimizing the execution of a workload to occur during those time periods when cleaner sources of power are available, emphasizing both spatial and temporal optimization. [3].

At the heart of the green software delivery standards are the Software Carbon Intensity (SCI) specifications, as promoted by the Green Software Foundation (2022). This industry led initiative provides a standardized approach to measure and

report the carbon footprint of various software artifacts, regardless of the deployment context. The SCI standard is critical in ensuring organizations are held accountable through auditing and communicating sustainability metrics effectively[4].

From the industry perspective, the CGI UK company has developed the Signal Sense platform that can be used to add carbon data APIs to the existing DevOps processes. In addition, the company has piloted the product in the UK government cloud environment to demonstrate the feasibility of the product in reducing the carbon footprint through dynamic rerouting based on real time grid conditions. [5].

Some related studies have investigated carbon sensitive job batching based on data based method selection. Bostandoost et al. (2025) proposed a system that selects the best scheduling method in real time based on the load of the task and the forecasted level of emission. This method increases the reduction of emission by over 8% compared to constant methods. This is based on the fact that flexibility and real time data analysis can make a real difference.[6].

In addition, the focus of the research into sustainability of the design of software includes the optimization of resource use both at the application level and at the level of the supporting infrastructure. The research into energy efficient algorithm's, programming techniques that are power aware, and advances in virtualization, shows that the way the software is designed has a major impact on the overall energy consumption; this complements "runtime" carbon aware scheduling solutions.

New studies are considering how generative artificial intelligence (AI) technology can be combined with global sustainability objectives in an effort to automate code optimization. The findings demonstrate that generative AI could reduce energy consumption of application executions by approximately 25% and accelerate execution speeds, presenting opportunities to create "greener" software than currently produced through operational monitoring.

In conclusion, the current state of green CI/CD research is an ever evolving, interdisciplinary body of work that combines the concepts of carbon aware scheduling, standardization, cloud infrastructure optimization, and intelligent software development. This provides a strong foundation for the development and implementation of green software engineering solutions that meet ecological and business needs [7].

The analysis of existing literature reveals that for sustainable software delivery pipelines, there are three key components which include real time environmental data integration and adaptive scheduling algorithms and standard measurement frameworks. The project of Hybrid Green CI/CD Framework comes in this context, which aims to take these trends forward by providing a portable solution for real time carbon aware DevOps

V. RESEARCH GAP

There are still a number of research gaps and problems in the foundations of Green CI/CD pipeline scheduling and carbon aware software delivery that limit the practical impact

and scalability of existing methods, despite advances that would allow for greater efficiency. Closing these gaps will be essential for the transition of Green CI/CD pipeline scheduling and carbon aware software delivery from an experimental platform to an accepted industrial standard.

One significant gap in the research is the heterogeneity and variability in the data related to carbon intensity. Although there are many projects that utilize the grid carbon data APIs, the quality and timeliness of the data differ from region to region. This affects the accuracy of the scheduling decisions, which may be suboptimal and may not be able to take full advantage of the reduction possibilities. The research has generally assumed that the data is reliable and stable, which is not the case in the real world[8].

The second barrier to implementing fully automated continuous delivery is achieving a balance between environmental sustainability and business objectives. A large number of scheduling algorithms focus on carbon reduction but do not have the capability to address issues such as emergency releases, tight deadlines or fluctuating workloads. As a result, the efficiency of pipelines will be compromised because manual overrides will be required to achieve complete automation or to achieve successful implementation of continuous delivery practices.

Another pressing gap is the lack of integration between hardware power consumption metrics and carbon intensity data. Most models focus on the external carbon footprint of the grid but do not take into account the internal variations in power consumption depending on the server type, virtualization stack, and container orchestration tools. This needs cross layer telemetry integration, which is complex and less explored.

In addition, there have been few studies on how to successfully scale and interoperate green CI/CD frameworks across various organizations and platforms, as most current solutions are focused on demonstrating the possibility of doing so or are closely tied to only one software tool (like GitHub Actions or Kubernetes). In order to make it easier for green CI/CD frameworks to be adopted within heterogeneous environments, there needs to be a generalizable and extensible architecture available so that they can be applied without excessive modification or operational burden.

The user adoption barrier and organizational readiness are yet another area where there has been a research gap. Carbon aware scheduling integration requires changes in the way the team works and the way the developer thinks. Most research has not provided in depth analysis in this area.

Standardizing and verifying sustainability metrics continue to challenge. The ability to audit emissions at the pipeline stage has advanced somewhat through the establishment of software carbon intensity standards; however, high confidence in these measurements is still difficult to achieve due to many factors impacting the ability to make this assessment. As a result, reliable measurement, reporting, and incentive mechanisms will require significant theoretical and experimental work to develop trust among users and for regulators to certify.

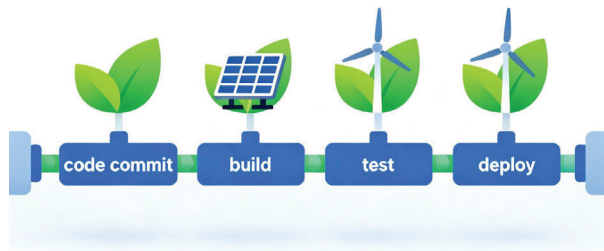


Fig. 2. CI CD pipeline.

As a result of this research gap analysis, it becomes evident that while the fundamentals of green CI/CD scheduling, complexity, scalability, as well as user adoption, are all covered, standardization of metrics remains an area of active research, which requires innovation to achieve sustainable SD.

VI. CONCLUSION AND FUTURE WORK

As a result of the expanding ecological footprint of software development and the deployment of applications, there is a greater demand for the integration of sustainability into DevOps. From this research, we know that incorporating carbon awareness into CI/CD (continuous integration/deployment) is possible by utilizing real time data about carbon emissions to provide dynamic scheduling and enforcement to lower the carbon footprint of code delivery while maintaining fast and reliable delivery of code. By implementing live carbon intensity data within the Hybrid Green CI/CD Framework, it supports a scalable model for developing sustainable software in a time when sustainability has come to the forefront.

One of the key takeaways from this research is that incorporating environmental metrics into software automation workflows fundamentally alters the traditional definition of DevOps. Traditionally, DevOps referred to either the speed of automation or the stability of deployment, but it is now an intelligent and contextually relevant participant in global carbon reduction efforts, which requires not just technology but also culture change in engineering groups.

Several operational challenges related to the issues of data variability, human factors, and hardware layer integration have been identified in the study, which point to the complex task of achieving complete green DevOps adoption. Future work will be focused on resolving these challenges by working on better data fusion techniques to improve the accuracy of carbon intensity prediction, implementing cross layer telemetry

for hardware aware emission computation, and security for external data dependencies.

The possibilities for machine learning and artificial intelligence to potentially provide additional optimization capabilities for carbon aware scheduling are encouraging. For example predictively modeled forecasts of the carbon output of the grid, workloads, and available resources may enable pipelines to run with greater detail, and zero lag time. Use of reinforcement learning techniques to allocate workloads and select tasks may yield adaptive policies for use under ever changing grid conditions and business needs without requiring human input, significantly improving sustainability.

However, scalability and the ability to work well with other tools in the Continuous Integration and Continuous Delivery (CI/CD) pipeline and the cloud are still important. Future research will involve the development of the components and APIs of the framework so that they can be integrated with other DevOps tools such as Jenkins, GitLab, and Azure DevOps. This will ensure the integration and adoption of carbon awareness.

Organizational change management is important in addition to the technological advances we already have today. Helping developers, managers, and other stakeholders see an accurate view of pipeline emissions will require creating intuitive dashboards and audit trails. This will help encourage changed behaviours. Putting in place sustainability KPIs for use in performance evaluations, as well as including them during project planning, can help develop green software engineering as an organizational culture value.

As the framework develops, it will also pursue compliance with regulations and laws along with certifications in different areas of compliance for sustainability. Governments and organizations are creating new standards for how to report emissions associated with digital businesses. As a result, providing auditor ready carbon data will be important. One example is that organizations will benefit from having compliance tools assisting them to meet the new standards set by the Green Software Foundation via the Software Carbon Intensity specification.

In conclusion, this work is a milestone in the way towards greener software deployment, since it is a concrete realization of the idea of carbon aware automation in CI/CD pipelines. The future is bright for green software development, since very soon software development will be enabled by these technologies to make a significant impact on the environment without compromising software development speed, software quality, and software user satisfaction.

REFERENCES

- [1] S. Patel *et al.*, "Carbon Aware Continuous Integration: Reducing Emissions Without Sacrificing Performance," *Environmental Modeling and Software Journal*, 2025. [Online]. Available: <https://lorojournals.com/index.php/emsj/article/view/1581>
- [2] S. Freed *et al.*, "Principles and Challenges in Green Software Engineering," *Green Software Engineering Survey*, 2023.
- [3] H. Claßen, J. Thierfeldt, J. Tochman Szewc, P. Wiesner, and O. Kao, "Carbon Awareness in CI/CD," in *ICSOC Workshops*, Springer, 2023. [Online]. Available: <https://arxiv.org/abs/2310.18718>

- [4] M. Pinnareddy *et al.*, "Carbon Conscious Scheduling in Kubernetes to Cut Energy Use and Emissions," *International Journal of Computational and Experimental Science and Engineering*, 2025.
- [5] Green Software Foundation, "Software Carbon Intensity (SCI) Specification," 2022. [Online]. Available: <https://greensoftware.foundation/>
- [6] CGI UK, "SignalSense – Intelligence as a Service," *UK Government G-Cloud Tech Reports*, 2024.
- [7] M. Bostandoost *et al.*, "Data driven Algorithm Selection for Carbon Aware Scheduling," *Journal of Sustainable Computing*, 2025.
- [8] N. Ashtekar, "Is Your CI/CD Pipeline Secretly Wasting Energy? Here's How to Fix It," *LinkedIn Article*, 2025. [Online]. Available: <https://www.linkedin.com/pulse/your-cicd-pipeline-secretly-wasting-energy-heres-how-fix-ashtekar-nhckf>
- [9] D. Lopes *et al.*, "Carbon Aware Build & Test Scheduling for Large Monorepos," *Well Testing Journal*, 2025.
- [10] Green Software Foundation, "Advances in Green Software Engineering," *GSF COP27 Report*, 2022.
- [11] M. Alnuaimi, "Energy Efficient Algorithms for Sustainable Software Delivery," *International Journal of Computer Engineering*, 2025.