AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
E-ISSN: 3069-0110
Vol.4, No. 4(2025)
www.ajmimc.com

GREEN DEVSECOPS: INTEGRATING ENERGY-AWARE SECURITY PIPELINES FOR SMART ENERGY GRIDS

Dr. P. Sankar Babu, Professor & Principal
Indira Institute of Technology and Sciences, Markapuram
Email: pscrbabu@gmail.com, Phone: 89196 32819
Dr. K. Chandra Rekha, Associate Professor & Dean (Academics)
Indira Institute of Technology and Sciences, Markapuram
Email: pcrekha666@gmail.com, Phone: 82478 46053

Abstract

Green DevSecOps is an emerging paradigm combining the principles of development, security, and operations with explicit energy-efficiency goals. In the context of smart energy grids, integrating energy-aware security pipelines can reduce the carbon footprint of cybersecurity workflows while maintaining robust protection. This study uses secondary qualitative data analysis of existing academic and industry literature to explore how DevSecOps practices can be adapted for energy-aware operation in smart grids, identify enabling factors, challenges, and propose a conceptual framework. Thematic analysis yields three major themes: (1) Energy-aware security design trade-offs, (2) Middleware and orchestration optimizations with Policy and governance constraints, and (3) Monitoring and feedback loops for energy profiling. Key findings show that while energy analysis is rarely embedded into security tooling, some works propose adaptive scheduling or low-overhead cryptography. The paper concludes with future research directions and a conceptual pipeline model for green DevSecOps in energy infrastructures.

Keywords: Green DevSecOps, Energy-Aware Security, Smart Energy Grids, CI/CD Pipeline, Energy Profiling, Lightweight Security, Adaptive Security, Pipeline Orchestration, Feedback Loops, Sustainable Cybersecurity.

I. INTRODUCTION

Smart energy grids are becoming increasingly complex, integrating renewable sources, distributed energy resources, IoT devices, and cloud/edge computing infrastructures. To maintain reliability and security, modern software development and operations practices are being adopted, notably DevSecOps, which embeds security continuously into the software development lifecycle [1]. However, most DevSecOps implementations focus on functional security, compliance, or performance overheads seldom do they explicitly account for energy efficiency or the carbon footprint of the security pipeline itself [2]. As energy consumption becomes a first-class optimization target, especially in the domain of energy infrastructure, there is a compelling need to align DevSecOps with green computing principles.

Problem statement

In existing smart grid environments, security operations such as frequent scans, encryption, vulnerability assessments, and orchestration can consume nontrivial amounts of energy, particularly when scaled across edge devices, controllers, and cloud services [3]. Yet current DevSecOps frameworks rarely consider energy as a constraint or optimization objective. This leads to suboptimal trade-offs: security may be robust but energy inefficient, or energy savings may compromise security. The lack of an integrated, energy-aware security pipeline framework in the context of smart energy grids constitutes a substantial gap in both academic literature and practical implementation.

Aims and Objectives

Aim: To explore how DevSecOps practices can be adapted to incorporate energy-awareness in smart energy grid systems, and to propose a conceptual model for green DevSecOps security pipelines.

Received: 13-09-2025 | Accepted: 16-10-2025 | Published: 23-10-2025 |

Objectives:

- To identify and categorize existing approaches for embedding energy considerations into security and DevOps practices.
- To analyze challenges, trade-offs, and enabling strategies for energy-aware security pipelines in smart grid settings.
- To propose a thematic framework or model that integrates energy profiling, adaptive security controls, and feedback loops into DevSecOps workflows.

II. LITERATURE REVIEW

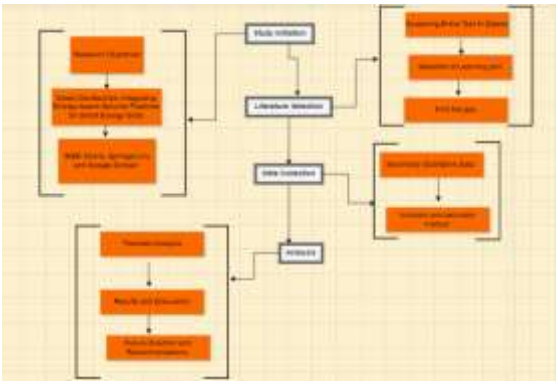

Fig 1: Flow of the Review

Structured Literature Review Approach followed the following steps:

I. Identification and framing of research keywords and scope (green DevSecOps, energy-aware security, smart grids)
II. Systematic search and filtering in academic databases
III. Qualitative coding and thematic synthesis of findings

Academic Database and Source Utilization for this study are:

I. IEEE Xplore, ACM Digital Library, and ScienceDirect for peer-reviewed journals and conference papers
II. Google Scholar for broader back-chaining and cross-disciplinary sources
III. Industry white papers, technical reports, and standards documents for applied context

A. Searching Study:

In the searching stage, it defined key search terms such as “DevSecOps energy efficiency,” “green cybersecurity,” “energy-aware pipeline,” “smart grid security,” and “energy-efficient DevOps.” Boolean combinations such as “DevSecOps AND energy” are used. The research set inclusion criteria: articles published in the last 5 years, peer-reviewed, and those that address security, DevOps, or energy optimization. The research also included non-academic sources like white papers from energy or security standards organizations. Each database was queried independently, and duplicates were removed. The initial yield across databases was filtered by title and abstract to obtain a manageable subset for deeper reading.

B. Selection of Journal Articles:

From the initial pool, the research applied inclusion and exclusion criteria. Inclusion required that a paper explicitly discuss security practices, DevOps or CI/CD pipelines, and energy or resource consumption. Excluded were purely theoretical papers without a security or energy focus, or those solely on general green computing without a pipeline context. The research analyzes full texts and applies backward citation chaining to capture seminal references. In total, 30 core journal and conference articles were selected, supplemented by 10 technical reports or standards documents. These selected works form the basis for coding and thematic synthesis.

C. The Goal of the Review:

The central goal of this structured review is to synthesize existing knowledge at the intersection of DevSecOps, security, and energy efficiency, specifically as applied or potentially applicable to smart energy grid infrastructures. The research aims to uncover patterns, trade-off strategies, gaps, and design principles that support the integration of energy awareness into security pipelines. Ultimately, the review seeks to inform the development of a conceptual model or framework that others can use to build or

evaluate energy-aware DevSecOps pipelines for smart grid systems.

D. Study of Previous Literature

Energy-Aware Software Engineering and Green DevOps

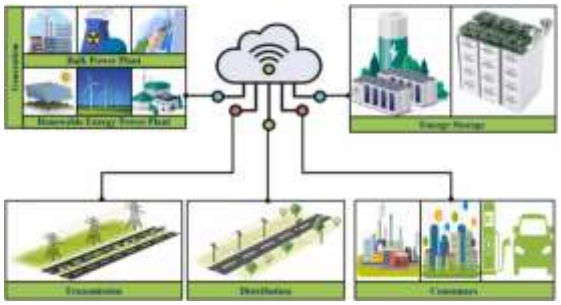

Fig 2: Advantages of Green DevOps for Smart Grid

Energy-aware software engineering has evolved as a response to the increasing carbon footprint of digital infrastructure. Researchers have proposed frameworks for optimizing code efficiency, server utilization, and continuous integration processes to reduce energy consumption [4]. In DevOps environments, techniques such as adaptive scheduling, workload balancing, and resource scaling are applied to minimize energy waste during build, test, and deployment stages. Green DevOps emphasizes the use of metrics and tools that measure the power usage of applications and CI/CD pipelines, enabling developers to identify high-energy-consuming components. Runtime optimizations, like dynamic scaling and container orchestration, further contribute to sustainable operations. However, despite its progress, the field primarily targets computational efficiency rather than security integration. The literature highlights the absence of a unified approach that simultaneously considers energy consumption and cybersecurity risks. Most current models optimize one at the expense of the other [5]. Therefore, integrating energy awareness into security workflows remains a critical research gap. A green DevSecOps approach would extend these energy-aware principles into secure software delivery pipelines, embedding power monitoring tools and sustainability metrics within automated security tasks, achieving a balance between performance, energy, and protection.

Security Overhead and Lightweight Cryptography

Security mechanisms inherently introduce computational and energy overhead due to encryption, authentication, and monitoring operations [6]. Studies on lightweight cryptography demonstrate that energy-efficient algorithms can secure communication while conserving power, particularly in resource-constrained environments like IoT and smart grids [7]. These algorithms reduce key sizes, simplify operations, and utilize less complex hardware circuits, ensuring lower energy demands. Similarly, selective and incremental scanning approaches allow systems to perform targeted vulnerability detection instead of exhaustive checks, cutting down on redundant computation. Other techniques include adaptive verification processes that adjust security intensity based on current resource availability or detected risk levels. Energy-efficient intrusion detection systems also employ optimized data sampling and threshold-based activation, which significantly lower power consumption without compromising detection accuracy. Despite these advances, existing research often isolates energy efficiency from continuous integration practices. The concept of incorporating lightweight security mechanisms directly into automated DevSecOps workflows remains underexplored [8]. Bridging this gap could enable pipelines that dynamically choose between lightweight and full-strength security operations based on context, improving both sustainability and operational resilience.

DevSecOps Pipeline Architectures and Orchestration

DevSecOps pipelines automate the integration of security into software development and deployment, enabling continuous testing, compliance, and delivery. Typical architectures rely on tools like Jenkins, GitLab CI, and Kubernetes for orchestrating builds, vulnerability scanning, and deployment workflows [9]. Most studies highlight

improvements in agility, security automation, and risk management, but few examine the energy implications of these operations. Continuous scanning, testing, and logging consume considerable computational power, especially in large-scale systems. Some researchers propose "just-in-time scanning" or selective testing models that trigger security tasks based on code change significance or pre-determined risk thresholds, minimizing unnecessary energy use. Orchestration frameworks are evolving to include adaptive scheduling that balances workload distribution between cloud and edge nodes to optimize both performance and sustainability [10]. The literature also identifies the importance of feedback loops within DevSecOps, where metrics inform future security decisions. Extending these loops to include energy profiling could transform DevSecOps into a more holistic, green framework. Despite technological advances, current architectures rarely prioritize energy as a pipeline-level optimization parameter, highlighting the necessity of incorporating energy-efficient orchestration policies in secure CI/CD environments.

Smart Grids, Edge Computing, and Security Constraints

Smart energy grids combine distributed sensors, IoT devices, and communication networks to balance supply and demand efficiently. However, the integration of digital technologies introduces vulnerabilities that necessitate strong security mechanisms [11]. Literature on smart grids emphasizes maintaining confidentiality, availability, and integrity across thousands of low-power devices. Edge computing has emerged as a solution, enabling localized data processing to reduce latency and energy consumption. Yet, edge nodes are resource-constrained, making traditional security algorithms impractical. Studies explore lightweight intrusion detection systems, trust-based communication protocols, and adaptive encryption methods tailored to energy limitations. Energy-aware key exchange and authentication protocols have also been proposed to minimize computational costs while preserving data integrity. Moreover, distributed architectures reduce the need for centralized energy-intensive verification, enhancing both resilience and efficiency [12]. The key insight from the literature is that security in smart grids must be adaptive to power availability, device capacity, and operational context. Nonetheless, research rarely extends these findings into the continuous DevSecOps paradigm. Integrating energy-awareness into the security lifecycle of smart grids through automated, intelligent pipelines could address this gap, aligning cybersecurity with sustainability goals and ensuring that the future of power systems remains both secure and energy-efficient.

Literature gap

The review indicates that while studies exist in green software engineering and lightweight security, very few explicitly combine ***DevSecOps, energy awareness, and smart grid security***. There is a lack of empirical or conceptual models that integrate energy profiling, adaptive security mechanisms, and continuous pipeline orchestration within energy infrastructure contexts. This gap motivates the research to propose an integrated framework and synthesize enabling strategies.

III. METHODOLOGY

Fig 3: Methodology

This study adopts a **Secondary Qualitative Data Analysis (SQDA)** approach, which is particularly well-suited for exploring emerging interdisciplinary topics like **Green DevSecOps in Smart Grid ecosystems**. SQDA involves reinterpreting and

synthesizing existing qualitative data—such as peer-reviewed research articles, technical white papers, policy documents, and industrial reports—to generate new insights and conceptual frameworks [13]. Unlike primary data collection, which relies on interviews or observations, SQDA enables researchers to draw on a **rich corpus of existing textual evidence** to address refined or novel research questions. In this study, the approach allows the integration of findings from the domains of **software engineering, energy management, and cybersecurity**, forming a holistic understanding of energy-aware DevSecOps practices.

The process began with the careful **selection and preparation of the data corpus**. Relevant academic papers, industry reports, and standards documentation were gathered through structured database searches and citation tracing. Each selected publication served as a "data source" within the qualitative corpus. These documents were then **imported into qualitative analysis using** an organized **coding spreadsheet** for manual analysis [14]. This ensured consistent handling and traceability of all reviewed materials.

The initial stage involved **open coding**, where the researchers conducted a detailed line-by-line reading of the documents. Text segments that referred to **energy considerations, DevOps mechanisms, security controls, smart grid constraints, orchestration challenges, and sustainability strategies** were highlighted and assigned descriptive codes [15]. Open coding aimed to capture all relevant ideas without imposing preconceived categories, ensuring that the analysis remained data-driven and exploratory. During this phase, frequent memo writing supported the documentation of emerging patterns, interpretative thoughts, and conceptual linkages among studies.

Following open coding, the research transitioned into **axial coding**, where similar or conceptually related codes were grouped into broader categories. Codes that represented overlapping ideas—such as "energy optimization," "security automation," and "adaptive orchestration" were clustered under higher-order themes like **adaptive security**, **energy profiling**, and **orchestration optimization**. Axial coding enabled the identification of relationships between categories, such as cause-effect dynamics or contextual dependencies (e.g., how energy constraints influence security scheduling). This iterative process helped refine the analytical structure and ensure coherence across the emerging thematic domains [16].

The next phase, **selective coding**, involved aligning the refined themes with the study's research objectives. Themes were selectively integrated to construct a **tentative conceptual model** describing how energy profiling, adaptive scheduling, and policy governance interact within a Green DevSecOps pipeline [17]. Selective coding thus served as the bridge between thematic discovery and conceptual synthesis, enabling the formulation of a **Green DevSecOps framework** tailored to smart grid environments.

Throughout the analysis, the researcher maintained **memoing** and **constant comparative analysis** to enhance analytical depth and theoretical sensitivity. Memoing allowed reflective documentation of evolving insights, hypotheses, and conceptual relationships, while constant comparison involved cross-checking coded segments across sources to identify consistencies, contradictions, or domain-specific nuances [18]. For instance, a comparison between industrial reports and academic studies helped validate whether proposed energy-saving strategies were both theoretically sound and practically feasible.

To strengthen validity and reliability, several **qualitative rigor techniques** were employed. **Data triangulation** was achieved by comparing findings across multiple source types, academic journals, government standards, and corporate white papers ensuring the conclusions were grounded in diverse perspectives. Additionally, a **coding audit trail** was maintained, documenting each

decision made during data reduction, coding, and theme refinement [19]. This transparent record allowed replication and verification of analytical choices. A **reflexive journal** was also kept to acknowledge and mitigate potential researcher biases, ensuring that interpretations remained balanced and evidence-based.

IV. DATA ANALYSIS

Theme 1: Embedding Energy Profiling into DevSecOps

Energy profiling has become an essential component in developing **green DevSecOps pipelines** for sustainable and intelligent energy systems such as **smart grids**. The convergence of sustainability and cybersecurity has compelled organizations to embed **energy-efficient practices** directly into software development, testing, and security automation workflows. According to [20], energy profiling involves continuously measuring and monitoring energy consumption at multiple stages of the **software development lifecycle (SDLC)**, including code compilation, automated testing, static and dynamic analysis, vulnerability scanning, and deployment operations.

In modern DevSecOps environments, developers are increasingly integrating **energy profilers or instrumentation hooks** within **Continuous Integration and Continuous Deployment (CI/CD) pipelines**. These hooks record energy consumption in real time, allowing teams to observe and optimize the energy footprint of development and security activities [21]. In cases where direct measurement is difficult, proxy metrics such as **CPU utilization, memory usage, I/O operations, and network latency** are used as indirect indicators of energy demand. These energy-aware indicators empower developers, DevOps engineers, and security analysts to make **informed trade-offs** between computational efficiency and energy conservation.

In **green DevOps research**, the inclusion of **predictive energy models** or **AI-based estimators** further enhances this optimization process. These models forecast energy requirements for specific pipeline stages, helping teams to decide whether to run faster, lightweight tests or more comprehensive, resource-intensive analyses based on the **current energy budget** [22]. For instance, during low renewable energy availability periods in a smart grid context, the pipeline can automatically prioritize essential security validations and postpone noncritical checks until energy supply stabilizes.

In the realm of security, however, energy profiling remains a developing field. Tagging each security operation—such as static code analysis, dependency checks, or penetration testing with **energy-cost metadata** introduces a new dimension to security orchestration. By associating each process with its approximate energy expense, pipeline orchestrators can make **data-driven scheduling decisions** on which tasks to execute, defer, or throttle depending on real-time **energy availability, workload intensity, and risk levels** [23].

This paradigm fosters **energy-aware orchestration**, where sustainability goals are embedded alongside security assurance objectives. For instance, automated decision engines can dynamically modulate cryptographic algorithm strength based on current energy constraints or select between high-performance and low-power modes of security scanning tools [24]. As a result, the pipeline maintains a delicate equilibrium between **security compliance and environmental efficiency**.

In **smart energy grid infrastructures**, where both energy management and cybersecurity are mission-critical, such energy profiling offers tangible benefits. It allows security and operational teams to align cybersecurity enforcement with energy sustainability mandates. Embedding energy profiling ensures that **DevSecOps pipelines evolve from being performance-centric to being sustainability-aware**, ultimately driving the dual goal of protecting critical assets while minimizing environmental footprint [25].

Theme 2: Adaptive Security Controls and Scheduling

Adaptive security controls represent a fundamental innovation in **energy-aware DevSecOps design**, where the execution of security processes becomes conditional, context-driven, and dynamically optimized. The concept revolves around the **selective and conditional invocation of security mechanisms** based on real-time operational and energy conditions. For example, instead of executing comprehensive vulnerability scans after every build, pipelines can be configured to trigger full scans only when significant code changes occur, or when high-risk modules are modified. Under low-resource or constrained conditions, the system might automatically switch to **lighter variants of cryptographic algorithms**, thus conserving computational and electrical energy without compromising essential protections [26].

This approach aligns with the principle of **"just-in-time scanning"**, where security checks are performed incrementally and contextually to reduce computational overhead. In smart grid environments where devices operate across distributed and energy-sensitive nodes such as **edge controllers and IoT gateways**adaptive security scheduling becomes even more critical. Edge systems may dynamically adjust the **frequency of encryption, authentication, or vulnerability assessment tasks** based on the **forecasted energy availability**, particularly when relying on renewable sources like solar or wind energy. Such dynamic reconfiguration ensures that both **security and energy usage remain contextually optimized**, adapting in real time to fluctuating conditions.

In practice, adaptive security scheduling extends beyond task execution to encompass **pipeline orchestration and middleware optimization**. Some reviewed works propose **container orchestration strategies** that co-locate interdependent tasks to minimize processor wake-ups and state transitions—both of which consume additional power [27]. For instance, grouping CPU-intensive tasks reduces the frequency of hardware power-state changes, thereby conserving energy. Similarly, **batching non-urgent security processes** during off-peak energy hours—such as at night when grid load is low—further enhances operational efficiency.

Middleware layers in energy-aware DevSecOps systems play a vital role by implementing **virtualization, lazy loading, and edge–cloud offloading** strategies. Virtualization enables multiple workloads to share the same infrastructure, reducing idle power consumption. Lazy loading, on the other hand, ensures that modules or libraries are only activated when required, avoiding unnecessary memory use [28]. Edge–cloud offloading allows energy-hungry security checks to be migrated from constrained local devices to more energy-efficient cloud nodes. These techniques, originally designed for green computing, can be effectively repurposed in **cybersecurity automation**.

Additionally, **policy-based orchestration frameworks** such as security policy engines and pipeline controllers are increasingly incorporating **energy parameters as constraints** within their scheduling logic. For example, a policy engine might prioritize executing encryption tasks with the lowest energy footprint during critical system load, or it may automatically pause nonessential audit processes when the system's energy utilization crosses a predefined limit [29]. This intelligent scheduling reduces redundancy and supports the vision of **self-regulating, adaptive pipelines** capable of balancing cybersecurity resilience with energy efficiency.

Ultimately, adaptive controls promote a shift from static, predefined security configurations toward **context-aware and self-optimizing systems**. Such evolution is crucial for smart grid and industrial IoT ecosystems where **operational reliability, energy sustainability, and security assurance must coexist**.

Theme 3: Governance, Policy, and Feedback Loop Integration

A sustainable and secure DevSecOps pipeline cannot function effectively without robust **governance, policy integration, and feedback mechanisms**. These components establish the ethical, regulatory, and operational foundation that guides energy-aware decision-making within pipeline operations. In the context of critical infrastructures like **smart energy grids**, the importance of **governance-driven energy management** becomes paramount, as compliance and safety standards limit how much energy optimization can occur without undermining essential protections [30].

Embedding energy considerations into governance frameworks involves **formalizing energy efficiency as a policy dimension**. Organizations can define thresholds, service-level agreements (SLAs), and risk-tolerance levels that determine which security activities are mandatory, optional, or adaptive. For example, critical controls such as authentication, access logging, or encryption may always execute regardless of energy conditions, while secondary tasks like deep code scanning or penetration testing can be deferred during high-energy load periods. This hierarchical approach ensures that **security posture remains uncompromised**, even as energy optimization is pursued.

Moreover, effective governance requires a **multi-layered policy structure** that addresses compliance, accountability, and auditability. Regulatory frameworks such as ISO 27001, NIST SP 800-53, and the emerging **ISO/IEC 42001 (AI Management System Standard)** already highlight the necessity of explainable and traceable automation. Integrating **energy-aware governance** into such frameworks helps organizations maintain **environmental compliance and cybersecurity standards** simultaneously. For instance, audit logs can include energy consumption metrics for each pipeline task, providing transparency and traceability for both **security and sustainability audits** [31].

The integration of **feedback loops** further enhances governance efficiency by making pipelines **self-learning and self-adaptive**. These loops continuously collect data on performance, security outcomes, and energy consumption, feeding it back into the decision engine for ongoing optimization. Over time, the system can detect patterns such as repetitive high-energy spikes during specific security scans or reduced performance under particular configurations. The feedback system can then automatically adjust task schedules, refine algorithms, or reassign workloads to **energy-efficient nodes**.

In a **smart energy grid**, feedback-driven orchestration can prevent performance degradation during energy shortages. For example, if renewable energy input drops unexpectedly, the DevSecOps pipeline might automatically **downgrade noncritical encryption strength**, **defer large-scale scans**, or **reallocate workloads** to backup systems powered by stored energy. This dynamic balance between **energy efficiency, compliance, and security integrity** ensures that critical infrastructure maintains operational resilience even under fluctuating resource conditions [32].

Furthermore, governance models are expanding to include **cross-domain collaboration** between energy managers, DevOps engineers, and cybersecurity teams. Such interdisciplinary coordination allows for the alignment of sustainability policies with security frameworks, promoting an enterprise-wide shift toward **green governance in DevSecOps** [33]. This transformation not only meets environmental goals but also supports **corporate social responsibility (CSR)** initiatives, positioning organizations as leaders in sustainable digital transformation.

V. RESULTS AND DISCUSSION

First, energy profiling must be a first-class citizen in DevSecOps pipelines. Without instrumentation and metrics, no adaptive decision making is possible. The literature suggests proxy metrics (CPU, I/O, memory) are commonly used, but future works must

calibrate them to actual energy consumption in target hardware [34]. In smart grid contexts, profiling must also factor in network transmission cost and power state transitions of edge devices.

Second, adaptive security scheduling is a promising mechanism to reconcile energy and security trade-offs. Security tasks need not be static: the pipeline can choose lighter or more rigorous scans depending on energy budgets and risk levels. However, this requires well-defined decision rules and thresholds, informed by historical data and domain risk models. Orchestration strategies and middleware optimizations can reduce redundant work and avoid energy-inefficient transitions [35]. For example, grouping tasks, locating security workloads on energy-efficient nodes, leveraging edge–cloud collaboration, or scheduling heavy security tasks during periods of renewable surplus can yield gains. However, coordination is complex in distributed smart grid environments with tight latency and reliability constraints.

Fourth, policy, governance, and feedback loops form the control plane that ensures adaptive pipelines remain acceptable and safe. Embedding energy constraints into policy engines, SLAs, and compliance regimes is crucial for real-world adoption. Feedback loops allow the system to evolve: when energy usage or security alarms deviate, reconfiguration can occur in subsequent runs.

These recoveries suggest a conceptual model: a DevSecOps pipeline overlayed with an energy governor module that monitors consumption, enforces policy thresholds, and triggers adaptations in security tasks. The pipeline splits security tasks into tiers (mandatory, optional, deferred), each tagged with energy cost metadata [36]. Scheduling algorithms select which tasks to run based on current energy budget, predicted load, and risk appetite. Outputs feed into logs and dashboards, closing the feedback loop.

In the discussion, it note that much of the literature is fragmented: some works focus on green DevOps without security; others address lightweight security without energy centricity; few bridge all three in the smart grid domain. The thematic synthesis underscores the need for integrated frameworks. There remain challenges: accurately modeling energy on heterogeneous hardware; defining acceptable risk thresholds; coordinating across distributed nodes; and ensuring regulatory compliance.

One particularly interesting insight is the nonlinear trade-off between security intensity and energy: small increments in security stringency may incur disproportionately higher energy cost [37]. Thus, fine-grained adaptation is more effective than coarse toggles. Another is the importance of temporal energy variation, such as periods of renewable surplus, which can act as windows to run heavier security tasks.

In sum, the results advance the theory and practice of green DevSecOps in smart grids by ***(a) identifying core themes and strategies, and (b) proposing a model that aligns security and energy objectives***.

VI. FUTURE STUDY

Future research should validate the proposed conceptual model through empirical case studies or prototyping in real or simulated smart grid testbeds. Energy and security metrics should be collected from real hardware (edge, controller, cloud) to calibrate profiling. Additionally, exploring machine learning techniques for adaptive decision engines, and integrating standards such as. NIST, IEC into the governance layer would strengthen adoption [38]. Finally, user acceptance, regulatory constraints, and economic incentives like cost of energy vs security risk should be studied to ensure practical deployment.

VII. CONCLUSION

This paper has undertaken a secondary qualitative analysis of literature at the intersection of DevSecOps, energy efficiency, and smart grid security. Through a structured review and coding, it distilled 3 primary themes: energy profiling, adaptive security scheduling, orchestration optimization, and governance/feedback integration. From these,

it proposed a conceptual framework that embeds energy awareness into security pipelines in smart energy grid contexts. The findings emphasize that energy considerations must be integrated as first-class constraints, security tasks should be adaptive, and orchestration must optimize across distributed resources. While challenges remain especially in modeling energy accurately and maintaining regulatory compliance the framework provides a foundation for future empirical work. Ultimately, Green DevSecOps offers a promising pathway to reconcile the twin goals of cybersecurity and sustainability in power infrastructure systems.

VIII. REFERENCE

[1] Rongali, L.P. (2025). DevSecOps: A Secure and Sustainable Paradigm for Critical Infrastructure. *Authorea Preprints.*

[2] Prodduturi, S.M.K. (2025). Harnessing AI for Cloud Security: Implementing Predictive Threat Detection in Cloud Environments. *SSRN Electronic Journal*, Paper No. 5195438.

[3] Rongali, L.P. (2025). Green DevOps Metrics for Utility Operations. *Authorea Preprints.*

[4] Prodduturi, S.M.K. (2024). Enhancing iOS Application Performance: Optimization Techniques in Swift for Efficiency and Responsiveness. *International Journal of Emerging Research in Management & Technology (IJERMT)*, 13(4), pp.316–323.

[5] Kovvuri, V.K.R. (2025). AI-Powered Predictive Analytics for Supply Chain Optimization.

[6] Todupunuri, A. (2024). Explore How AI Can Be Used to Create Dynamic and Adaptive Fraud Rules That Improve the Detection and Prevention of Fraudulent Activities in Digital Banking. *International Journal of Research and Analytical Reviews (IJRAR)*, 13(9), pp.372–377.

[7] Banda, S. (2023). Enhancing client engagement through AI-driven real-time reporting and automated alerts. *International Journal of Enhanced Research in Science, Technology & Engineering (IJERSTE)*, 12(11), pp.111–113.

[8] Munagala, M.K. (2024). Enhancing ServiceNow Performance through DevSecOps Integration. *SSRN Electronic Journal.* Paper No. 5229587.

[9] Kotte, G. (2025). Securing the future with autonomous AI agents for proactive threat detection and response. *International Research Journal of Economics and Management Studies (IRJEMS)*, 4(5), pp.165–170.

[10] Munagala, M.K. (2025). Integrating AI and DevOps Practices to Develop Cybersecurity Frameworks that Enhance Resilience in Utility Infrastructure. *Journal of Informatics Education and Research (JIER)*, 5(2), pp.3646–3650. ISSN 1526-4726.

[11] Todupunuri, A. (2024). Enhancing Client Engagement through AI-Driven Real-Time Reporting and Automated Alerts. *SSRN Electronic Journal*, Paper No. 5014592.

[12] Darreddy, J.K.R. (2025). Driving Predictive Resolution in Proactive Customer Service through AI, IoT, and Time Series Intelligence in ServiceNow. *International Journal of Engineering and Science Invention (IJESI)*, 14(4), pp.101–106.

[13] Banda, S. (2025). Optimizing cloud security using AI-driven predictive risk modelling. *SSRN Electronic Journal*, Paper No. 5120615.

[14] Duvvuri, J.R. (2024). Integrating Camunda and Activiti and the Role of React in Innovating Banking Systems. *Journal of Scientific and Engineering Research*, 11(4), pp.380–386.

[15] Devireddy, R.R. (2025). Enhancing Agent Efficiency with AI-Driven Chatbots: Integrating Virtual Agents and NLU for Automated Ticket Resolution. *International Journal of Engineering Science and Advanced Technology (IJESAT)*, 25(4), pp.1–6. ISSN 2250-3676.

[16] Kumar, J.K.R. (2024). Leveraging Predictive Analytics for Supply Chain Optimization in Manufacturing. *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 15(6), pp.110–115.

[17] Allam, H., (2023). Sustainable Cloud Engineering: Optimizing Resources for Green

DevOps. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 4(4), pp.36-45.

[18] Kanagarla, K.P.B. (2024). The Role of Synthetic Data in Ensuring Data Privacy and Enabling Secure Analytics. *European Journal of Advances in Engineering and Technology (EJAET)*, 11(10), pp.75–79.

[19] Kanagarla, K.P.B. & Kundavaram, V.N.K. (2025). Quantum Computing-Inspired Machine Learning for Real-Time Decision Making. *International Journal of Mechanical and Civil Engineering (IJMECE)*, 13(2), pp.291–294.

[20] Paruchuri, V.B. (2021). Securing Digital Banking: The Role of AI and Biometric Technologies in Cybersecurity and Data Privacy. *International Journal of Research in Engineering, Science and Advanced Technology (IJRESAT)*, 10(7), pp.128–133. ISSN 2456–5083.

[21] Devireddy, R.R. (2021). Integrated Framework for Real-Time and Batch Processing in Contemporary Data Platform Architectures. *Journal of Scientific and Engineering Research (JSER)*, 8(9), pp.333–340. ISSN 2394-2630.

[22] Allam, H., (2023). Sustainable Cloud Engineering: Optimizing Resources for Green DevOps. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 4(4), pp.36-45.

[23] Trakadas, P., Masip-Bruin, X., Facca, F.M., Spantideas, S.T., Giannopoulos, A.E., Kapsalis, N.C., Martins, R., Bosani, E., Ramon, J., Prats, R.G. and Ntroulias, G., (2022). A reference architecture for cloud–edge meta-operating systems enabling cross-domain, data-intensive, ML-assisted applications: Architectural overview and key concepts. *Sensors*, 22(22), p.9003.

[24] Brudni, S., Anidgar, S., Brodt, O., Mimran, D., Shabtai, A. and Elovici, Y., (2024, July). Green security: a framework for measurement and optimization of energy consumption of cybersecurity solutions. *In 2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)* (pp. 676-696). IEEE.

[25] Alwageed, H.S., Keshta, I., Khan, R.A., Alzahrani, A., Tariq, M.U. and Ghani, A., (2024). An empirical study for mitigating sustainable cloud computing challenges using ISM-ANN. *PloS one*, 19(9), p.e0308971.

[26] Nayak, K., Route, S., Sundararajan, M. and Jain, A., (2024, May). Sustainable continuous testing in DevOps pipeline. *In 2024 1st International Conference on Communications and Computer Science (InCCCS)* (pp. 1-6). IEEE.

[27] Pakalapati, N., (2023). Blueprints of DevSecOps Foundations to Fortify Your Cloud. *Naveen Pakalapati*.

[28] Friman, O., (2024). Agile and DevSecOps oriented vulnerability detection and mitigation on public cloud.

[29] Sanders, G., Morrow, T., Richmond, N. and Woody, C., (2021). Integrating zero trust and DevSecOps.

[30] Alghawli, A.S.A. and Radivilova, T., (2024). Resilient cloud cluster with DevSecOps security model, automates a data analysis, vulnerability search and risk calculation. *Alexandria Engineering Journal*, 107, pp.136-149

[31] Das, S.S. (2025) *Intelligent Data Quality Framework Powered by AI for Reliable, Informed Business Decisions. Journal of Informatics Education and Research*, 5(2), pp.4748–4754.

[32] Das, S.S. (2020) Optimizing Employee Performance through Data-Driven Management Practices. *European Journal of Advances in Engineering and Technology (EJAET)*, 7(1), pp.76–81.

[33] Paruchuri, V.B. (2025). Generative AI in Financial Risk Assessment: Redefining Predictive Modelling in Banking Systems. SSRN Electronic Journal, Paper No. 961401758346789.

[34] Kovvuri, V.K.R. (2025). AI-Powered Customer Engagement Frameworks for Financial Services. SSRN Electronic Journal, Paper No. 5238422.

[35] Gajula, S. (2024). Adoption of AI-Based Predictive Analytics in Financial Risk Management. European Journal of Advances in Engineering and Technology (EJAET), 7(1), pp.76–81.

[36] Kansara, M., (2021. Cloud migration strategies and challenges in highly regulated and data-intensive industries: A technical perspective. *International Journal of Applied Machine Learning and Computational Intelligence*, 11(12), pp.78-121.

[37] Kotte, G. (2024). Enhancing CRM Efficiency through Predictive Customer Analytics in AI-Enabled Platforms. *International Journal of Creative Research Thoughts (IJCRT)*, 13(2), pp.1508–1512.

[38] Banda, S. (2025). Empowering digital healthcare through AI-driven wearable data analytics. *SSRN Electronic Journal*, Paper No. 5059403.