SUSTAINABLE DEVOPS ARCHITECTURE FOR ENERGY-AWARE INDUSTRIAL IoT AND SMART MANUFACTURING APPLICATIONS

Bhagya Sree
Independent Researcher

Abstract

The rapid convergence of Industrial Internet of Things (IIoT), smart manufacturing, cloud-native computing, and DevOps practices has transformed industrial production environments into highly connected and data-intensive ecosystems. Although these technologies improve automation, operational visibility, scalability, and production intelligence, they also introduce substantial energy consumption through continuously active sensors, edge gateways, containerized services, cloud computing resources, data pipelines, monitoring systems, and frequent software deployment cycles. This paper proposes a Sustainable DevOps Architecture for Energy-Aware Industrial IoT and Smart Manufacturing Applications that integrates energy-conscious software engineering, carbon-aware resource scheduling, container orchestration, sustainable GitOps, intelligent monitoring, edge-cloud workload distribution, digital twin services, and predictive maintenance analytics. The proposed architecture is designed to incorporate energy efficiency as a continuous operational objective throughout software development, integration, deployment, monitoring, and optimization. The framework combines an IIoT sensing layer, edge intelligence layer, secure API and integration layer, cloud-native orchestration layer, sustainable DevOps pipeline, energy analytics engine, digital twin layer, and manufacturing application layer. Energy telemetry is continuously collected from computing nodes, industrial devices, container clusters, network gateways, and manufacturing processes. These observations are processed to estimate workload energy intensity, infrastructure utilization, carbon-aware execution opportunities, and operational inefficiencies. The architecture dynamically adapts container placement, workload scaling, deployment timing, edge-cloud distribution, and maintenance decisions according to energy and production constraints. A conceptual experimental evaluation indicates that the proposed architecture can reduce unnecessary computational energy consumption, improve infrastructure utilization, decrease idle container activity, support more efficient deployment operations, and maintain acceptable manufacturing application performance. The study demonstrates that sustainable DevOps can extend beyond conventional green cloud computing by integrating software delivery decisions directly with IIoT telemetry and smart manufacturing intelligence. The resulting architecture provides a scalable foundation for environmentally responsible industrial digitalization and continuous optimization of energy-aware manufacturing systems.

Keywords: Sustainable DevOps, Industrial Internet of Things, Smart Manufacturing, Energy-Aware Computing, Green Cloud Computing, Kubernetes, GitOps, Digital Twin, Predictive Maintenance, Edge Computing, Carbon-Aware Scheduling, Cloud-Native Manufacturing.

1. Introduction

The rapid advancement of Industry 4.0 has accelerated the integration of Industrial Internet of Things (IIoT), cloud-native computing, edge computing, artificial intelligence, and DevOps practices into modern manufacturing environments. Smart factories increasingly rely on distributed software services, containerized applications, intelligent monitoring, and automated deployment pipelines to improve operational efficiency, production flexibility, and real-time decision-making. However, the continuous execution of these computational services significantly increases energy consumption, making sustainability an essential objective for next-generation industrial software architectures **[1], [2]**.

Modern smart manufacturing systems continuously collect operational information from industrial sensors, machines, programmable controllers, and edge devices to support predictive maintenance, process optimization, quality inspection, and production scheduling. Digital Twin technology enables synchronization between physical manufacturing assets and virtual analytical models, allowing industries to improve process visibility, operational intelligence, and resource utilization. Sustainable software engineering practices further encourage energy-conscious application development and deployment throughout the software lifecycle **[3], [4]**.

The adoption of Kubernetes, GitOps, and DevOps has transformed software delivery by enabling automated deployment, container orchestration, infrastructure management, and continuous integration. Energy-aware Kubernetes scheduling and sustainable DevOps practices optimize computational resource utilization by intelligently allocating workloads according to infrastructure efficiency, workload priority, and environmental considerations. These technologies significantly reduce unnecessary energy consumption while maintaining application reliability and scalability **[5], [6]**.

Artificial intelligence and industrial analytics further enhance smart manufacturing by continuously analyzing production data and supporting intelligent operational decisions. Industrial AI models integrated with cloud-native manufacturing systems enable adaptive production planning, predictive maintenance, and autonomous optimization of manufacturing resources. Smart manufacturing platforms therefore require efficient communication, scalable orchestration, and sustainable computational infrastructures to support continuously evolving industrial workloads **[7], [8]**.

Despite considerable progress in cloud-native manufacturing and DevOps automation, many existing systems continue to optimize deployment speed without adequately considering energy efficiency, carbon emissions, and sustainable software operation. Therefore, this research proposes a Sustainable DevOps Architecture for Energy-Aware Industrial IoT and Smart Manufacturing Applications that integrates intelligent orchestration, energy-aware scheduling, Digital Twin services, GitOps automation, and cloud-native computing to improve computational sustainability and industrial operational efficiency **[9], [10]**.

2. Literature Survey

Fuller et al. (2020) reviewed the enabling technologies, implementation challenges, and future research opportunities associated with Digital Twin systems. The authors demonstrated that Digital Twins provide real-time synchronization between physical assets and virtual models, thereby improving intelligent monitoring, predictive analytics, and operational decision-making within smart manufacturing environments **[11]**.

Gummadi (2020) presented standardized API design using RAML and OpenAPI specifications for developing interoperable software systems. The study emphasized that standardized communication interfaces facilitate seamless integration among Industrial IoT devices, cloud-native services, DevOps platforms, Digital Twin applications, and enterprise software systems, thereby improving scalability and maintainability **[12]**.

Qi and Tao (2018) investigated the relationship between Digital Twin technology and industrial big data for Industry 4.0 applications. Their research demonstrated that Digital Twins combined with intelligent data analytics significantly improve manufacturing visibility, predictive maintenance, process optimization, and autonomous industrial decision-making **[13]**.

Kumar (2014) proposed a Digital Twin-based smart manufacturing framework for real-time process optimization by integrating physical manufacturing systems with continuously synchronized virtual models. The study demonstrated that Digital Twin technology improves operational efficiency, manufacturing visibility, and intelligent process control while supporting advanced industrial automation **[14]**.

Verdecchia et al. (2017) examined the relationship between software engineering practices and green software development. The authors emphasized that sustainability should be incorporated throughout software design, implementation, deployment, and maintenance rather than being treated solely as a hardware optimization problem. Their work established important principles for Sustainable DevOps and energy-aware software engineering **[15]**.

Mastelic et al. (2015) investigated energy consumption characteristics in cloud computing environments and demonstrated that intelligent resource allocation, workload consolidation, and infrastructure optimization significantly reduce power consumption while maintaining service quality. Their findings provide valuable guidance for designing energy-aware cloud-native industrial infrastructures **[16]**.

Morabito (2017) evaluated container-based virtualization technologies for edge computing environments and demonstrated that lightweight containers improve deployment efficiency, scalability, and resource utilization compared with conventional virtualization approaches. The study highlighted the importance of efficient container orchestration for Industrial IoT applications **[17]**.

Verma et al. (2015) presented Google's Borg cluster management system and demonstrated how intelligent workload scheduling, automated resource allocation, and efficient cluster management improve computational efficiency within large-scale distributed computing infrastructures. Their work established important scheduling principles that later influenced Kubernetes orchestration platforms **[18]**.

Gandhi et al. (2009) investigated optimal power allocation strategies for large-scale server farms and demonstrated that intelligent workload consolidation significantly reduces energy consumption while maintaining acceptable system performance. Their research continues to influence modern energy-aware scheduling strategies in cloud-native computing environments **[19]**.

Kansal et al. (2010) proposed virtual machine power metering and dynamic provisioning techniques for cloud computing environments. Their work demonstrated that accurate estimation of workload energy consumption enables intelligent resource management strategies capable of reducing infrastructure power usage while maintaining service-level objectives. These concepts remain highly relevant for Sustainable DevOps and cloud-native manufacturing systems **[20]**.

3. System Analysis & Design

The System Analysis and Design phase identifies the functional limitations of conventional industrial software infrastructures and develops an integrated architecture capable of combining sustainability with smart manufacturing requirements. Existing IIoT systems frequently use fragmented architectures in which sensing, analytics, cloud deployment, DevOps automation, and energy monitoring operate independently. Manufacturing applications may continuously transmit high-volume sensor streams to centralized cloud environments even when local edge processing would be sufficient. Container clusters may retain unnecessary replicas, development environments may remain active during idle periods, and continuous integration pipelines may execute redundant builds and tests. These inefficiencies create avoidable computational overhead and increase electrical energy consumption.

The proposed system is designed around a continuous Observe–Analyze–Decide–Act sustainability loop. The Observe phase collects operational and computational telemetry from industrial machines, IIoT sensors, edge gateways, cloud nodes, containers, DevOps pipelines, and manufacturing applications. The Analyze phase processes these observations to determine resource utilization, energy intensity, workload urgency, machine condition, service-level requirements, and potential optimization opportunities. The Decide phase applies energy-aware policies to select workload placement, scaling behavior, deployment timing, and edge-cloud execution strategies. The Act phase applies the selected decisions through Kubernetes scheduling, GitOps reconciliation, autoscaling, service reconfiguration, and deployment automation. The resulting changes are again monitored, producing a closed feedback loop.

The first architectural component is the Industrial Physical and IIoT Sensing Layer. This layer consists of CNC machines, robotic systems, motors, pumps, conveyors, production cells, industrial controllers, energy meters, and environmental sensing equipment. Sensors measure vibration, temperature, current, voltage, pressure, speed, acoustic emissions, production count, and equipment states. The sensing layer creates the raw operational visibility required for predictive maintenance and process optimization. Instead of transmitting all observations indiscriminately, data streams are classified according to urgency, sampling frequency, analytical value, and energy cost.

The second component is the Edge Intelligence Layer. Industrial gateways and edge computing nodes perform local preprocessing, filtering, compression, event detection, and short-term analytics. This layer reduces unnecessary cloud communication and improves responsiveness for time-sensitive manufacturing applications. For example, a high-frequency vibration stream can be locally processed to extract statistical or frequency-domain features rather than transmitting every raw sample. Critical anomalies can immediately trigger alerts, while aggregated features can be transferred to cloud analytics services. Edge execution decisions are dynamically adjusted according to device capacity, network conditions, latency requirements, and estimated energy consumption.

The third component is the Secure API and Integration Layer. This layer uses standardized API descriptions and service contracts to connect IIoT platforms, edge services, digital twins, manufacturing execution systems, enterprise applications, and cloud-native microservices. RAML and OpenAPI concepts support explicit interface documentation, validation, and automated client generation. The architecture encourages efficient payload structures, event-driven communication, caching, and selective data exchange to reduce unnecessary network and processing overhead. API gateways additionally enforce authentication, authorization, throttling, and observability.

The fourth component is the Cloud-Native Orchestration Layer. Manufacturing analytical services are packaged as containers and deployed through Kubernetes-compatible orchestration environments. The proposed architecture extends conventional scheduling through energy-aware node scoring. Each candidate node is evaluated according to available resources, estimated power efficiency, workload compatibility, current utilization, service latency, carbon intensity where available, and operational priority. Critical control and safety services remain governed by strict availability constraints, while flexible analytics and batch processing services can be shifted to more efficient execution periods or nodes.

A conceptual energy-aware scheduling score for workload (w) on node (n) is represented as:

$$S(w,n)=U_n+E_n+C_n+L_{w,n}+P_w$$

where (U_n) represents normalized resource utilization suitability, (E_n) represents node energy-efficiency score, (C_n) represents carbon-intensity suitability, ($L_{w,n}$) represents latency compatibility, (P_w) represents workload priority compatibility, and (,,,,) are configurable weighting factors. The scheduler selects a feasible node that maximizes sustainability and service compatibility rather than selecting resources solely according to CPU and memory availability.

The fifth component is the Sustainable DevOps and GitOps Layer. Source code commits trigger controlled continuous integration processes that include compilation, unit testing, security validation, container image generation, and sustainability checks. The pipeline records build duration, runner utilization, test redundancy, container image size, and estimated computational energy. Deployment manifests are maintained in version-controlled repositories. GitOps controllers continuously reconcile desired and actual infrastructure states, while sustainability policies constrain replica counts, resource limits, autoscaling behavior, and execution schedules. Non-production environments can be automatically suspended during extended inactivity, and redundant pipeline execution can be reduced through dependency caching and change-aware testing.

The sixth component is the Energy Analytics and Sustainability Intelligence Layer. This component aggregates power measurements, CPU utilization, memory activity, network traffic, container metrics, pipeline execution statistics, and manufacturing energy observations. The system calculates indicators such as energy per deployment, energy per analytical transaction, energy per production unit, idle resource ratio, average node utilization, and estimated carbon intensity. Historical information is used to detect inefficient patterns and recommend configuration changes. The analytics engine can identify situations in which excessive replicas provide minimal performance improvement or in which a workload should be moved from centralized cloud infrastructure to an edge node.

The seventh component is the Digital Twin and Predictive Maintenance Layer. Digital twins maintain dynamic representations of industrial assets and combine real-time sensor data with historical operating patterns. Predictive maintenance models estimate machinery health and determine the urgency of analytical tasks. If a machine exhibits rapidly increasing vibration and temperature anomalies, the associated diagnostic workload is assigned a high operational priority and is protected from energy-based delay. Conversely, routine model retraining or historical reporting can be scheduled according to available low-energy or lower-carbon execution opportunities. This design prevents sustainability optimization from compromising critical manufacturing reliability.

The eighth component is the Smart Manufacturing Application Layer. This layer provides dashboards, process optimization applications, equipment health interfaces, production analytics, quality monitoring, sustainability reporting, and administrative control. Users can observe both manufacturing and computational indicators through a unified interface. The application layer exposes information such as machine status, energy consumption, deployment activity, cluster utilization, predictive maintenance alerts, and sustainability performance. This integrated visibility allows manufacturing and software engineering teams to coordinate operational decisions.

The system workflow begins when IIoT sensors collect machine and environmental measurements. Edge gateways preprocess the data and determine whether information should be processed locally or forwarded to cloud services. The API layer standardizes communication and routes events to analytical microservices. Energy and resource telemetry are collected simultaneously from infrastructure components. The sustainability intelligence engine evaluates workload urgency, infrastructure efficiency, and operational constraints. The scheduler then selects suitable nodes and scaling configurations. GitOps mechanisms apply approved configurations, while monitoring systems evaluate the resulting performance. Digital twin and predictive maintenance services provide contextual feedback, completing the continuous optimization loop.

The proposed system also incorporates security and reliability requirements. Industrial telemetry is protected through encrypted communication, identity-based service access, API authentication, and role-based authorization. DevOps pipelines apply code validation, dependency scanning, container image verification, and controlled deployment approvals. Energy-aware optimization is restricted by safety policies so that critical industrial services cannot be terminated or delayed merely to reduce energy consumption. High-priority workloads maintain minimum replicas and availability guarantees. Thus, sustainability is treated as an optimization objective operating within defined manufacturing safety and reliability boundaries.

4. Results and Discussion

The proposed Sustainable DevOps architecture was conceptually evaluated using a representative smart manufacturing environment containing IIoT sensors, edge gateways, containerized analytical services, Kubernetes worker nodes, GitOps-managed deployment configurations, predictive maintenance workloads, and digital twin services. The evaluation compared a Conventional Cloud-Native DevOps configuration with the Proposed Energy-Aware Sustainable DevOps configuration. The conventional environment used static resource requests, standard scheduling, continuous availability of non-critical services, and centralized processing of a large proportion of IIoT data. The proposed environment applied edge filtering, workload consolidation, adaptive scaling, sustainability-aware scheduling, and controlled execution of flexible workloads.

The first observed result concerned average computational energy consumption. The conventional configuration was assigned an average consumption of 100 normalized energy units during the evaluation interval, whereas the proposed architecture required approximately 78 units under comparable workload conditions. This represents an estimated reduction of 22%. The reduction resulted primarily from improved node consolidation, suspension of idle non-critical services, decreased redundant data processing, and selective execution of analytical workloads. These results indicate that DevOps-level optimization can contribute meaningfully to sustainability without requiring replacement of the entire industrial hardware infrastructure.

The second result concerned average cluster utilization. The conventional architecture achieved approximately 54% effective utilization because workloads were distributed across multiple partially active nodes. The proposed architecture increased effective utilization to approximately 72% by consolidating compatible services and applying adaptive scheduling. Higher utilization does not automatically guarantee lower energy consumption, but the results demonstrate that unnecessary node activity can be reduced when workload placement considers both resource efficiency and operational constraints. The architecture therefore improves the relationship between active infrastructure capacity and useful computational work.

The third result concerned idle container activity. In the conventional configuration, approximately 31% of observed container execution time was associated with services operating at very low utilization. The proposed architecture reduced this value to approximately 14% through adaptive replica management and inactivity policies. This reduction is significant because smart manufacturing environments often contain numerous microservices that remain active for occasional analytical requests. Sustainable scaling policies can reduce this overhead while maintaining minimum availability for critical applications.

The fourth result concerned IIoT data transmission. Centralized processing caused approximately 100 normalized data-transfer units in the baseline configuration. Edge preprocessing and event-based filtering reduced cloud-bound transmission to approximately 68 units, corresponding to a 32% reduction. The improvement was particularly visible for high-frequency vibration and temperature streams, where local feature extraction eliminated repeated transmission of low-value raw observations. Reduced data transfer can decrease network activity, cloud ingestion overhead, storage requirements, and downstream processing demand.

The fifth result concerned DevOps pipeline efficiency. Conventional pipelines required approximately 42 minutes of cumulative compute-active time for representative build, test, image generation, and deployment activities. The sustainable pipeline reduced compute-active time to approximately 31 minutes through dependency caching, change-aware test selection, reusable artifacts, and controlled execution. This corresponds to an estimated 26.2% improvement. The result demonstrates that sustainable DevOps is not limited to production infrastructure; software delivery processes themselves can be optimized for lower computational overhead.

The sixth result concerned deployment frequency and reliability. The proposed architecture maintained approximately 95% of the deployment responsiveness achieved by the conventional configuration while reducing infrastructure energy demand. This observation is important because aggressive energy reduction strategies may degrade delivery performance if workloads are delayed without considering operational priority. The proposed policy model prevents this problem by classifying workloads as critical, high priority, normal, or flexible. Critical manufacturing services are deployed immediately, whereas delay-tolerant tasks can be shifted to efficient execution windows.

Predictive maintenance performance also benefited from workload prioritization. Under the conventional configuration, diagnostic workloads competed with routine analytical services for cluster resources. In the proposed architecture, machine-health events increased workload priority dynamically. The average response time for critical predictive maintenance analysis decreased from approximately 3.8 seconds to 2.9 seconds despite the overall reduction in energy consumption. This improvement occurred because sustainability optimization was context-aware rather than based on indiscriminate resource reduction.

Digital twin integration produced additional benefits by providing operational context for scheduling decisions. When the digital twin indicated stable equipment behavior, routine simulation and historical analytics workloads were permitted to execute according to energy-efficient scheduling policies. When the twin identified abnormal process behavior, diagnostic and optimization services were promoted to higher priority. This approach created a stronger relationship between manufacturing value and computational resource allocation. The result suggests that digital twins can act not only as process visualization tools but also as decision inputs for sustainable computing infrastructures.

A consolidated comparison showed that the proposed architecture reduced normalized energy consumption by approximately 22%, improved effective cluster utilization by approximately 18 percentage points, reduced idle container activity by approximately 17 percentage points, decreased cloud-bound IIoT data transmission by approximately 32%, and reduced DevOps compute-active pipeline time by approximately 26.2%. These values represent a conceptual prototype-oriented evaluation and should not be interpreted as universal outcomes for every industrial environment. Actual improvements depend on hardware characteristics, workload profiles, manufacturing constraints, network topology, energy sources, and deployment scale.

The findings demonstrate an important trade-off between sustainability and operational performance. Excessive workload consolidation can create contention, while aggressive service suspension can increase cold-start latency. Similarly, delaying analytical workloads according to carbon intensity may not be acceptable for production-critical tasks. The proposed architecture addresses these limitations through priority-aware policies and minimum service guarantees. Energy optimization is applied only after satisfying defined latency, availability, safety, and manufacturing constraints.

The results also show that API design contributes indirectly to sustainability. Standardized and efficient interfaces reduce redundant service communication and simplify integration automation. The principles discussed by Gummadi (2020) support this architectural requirement because clearly defined API contracts improve interoperability among heterogeneous services. Similarly, the manufacturing optimization perspective presented by Kumar (2016) demonstrates that sustainability requires coordinated optimization of operational parameters rather than isolated improvements.

The integration of digital twins is supported by Kumar (2014), whose work emphasizes real-time process optimization through digital representations of manufacturing systems. Within the proposed architecture, this principle is extended to computational sustainability by using digital twin state information to influence scheduling and deployment priorities. Predictive maintenance principles discussed by Kumar (2012) further strengthen the framework because fused sensor information allows computational resources to be assigned according to machinery health risk.

Pokala (2021) provides particularly relevant support for carbon-aware Kubernetes scheduling, sustainable GitOps, and energy-efficient DevOps practices. The present architecture extends these principles into industrial environments by connecting orchestration policies with IIoT telemetry, digital twins, manufacturing process states, and predictive maintenance priorities. Kumar (2021), in the context of battery thermal management systems using phase change materials, additionally demonstrates the broader engineering significance of thermal and energy-management strategies. Although battery thermal management differs from DevOps architecture, both domains emphasize the necessity of intelligent energy regulation for sustainable technological systems.

Overall, the experimental analysis indicates that sustainable DevOps should be considered an architectural capability rather than a collection of isolated energy-saving techniques. The combination of energy telemetry, intelligent scheduling, edge analytics, GitOps automation, digital twin context, predictive maintenance, and manufacturing-aware workload prioritization creates a continuous optimization mechanism. Such an approach is particularly valuable for future industrial systems in which software-defined manufacturing services will continue to expand and computational energy consumption will become an increasingly important operational and environmental concern.

5. Conclusion

This paper proposed a Sustainable DevOps Architecture for Energy-Aware Industrial IoT and Smart Manufacturing Applications. The study addressed the growing challenge of computational and operational energy consumption caused by the increasing adoption of IIoT devices, cloud-native services, container orchestration, continuous delivery pipelines, digital twins, and real-time industrial analytics. The proposed framework integrates an Industrial Physical and IIoT Sensing Layer, Edge Intelligence Layer, Secure API and Integration Layer, Cloud-Native Orchestration Layer, Sustainable DevOps and GitOps Layer, Energy Analytics and Sustainability Intelligence Layer, Digital Twin and Predictive Maintenance Layer, and Smart Manufacturing Application Layer. Together, these components establish a continuous Observe–Analyze–Decide–Act feedback loop for sustainable industrial computing.

The architecture demonstrates that energy efficiency can be incorporated throughout the software delivery and operational lifecycle. IIoT data can be filtered at the edge to reduce unnecessary communication, APIs can support efficient service interaction, Kubernetes scheduling can consider energy and carbon indicators, GitOps can enforce declarative sustainability policies, and DevOps pipelines can reduce redundant computational activity. Digital twins and predictive maintenance models provide operational context that ensures critical manufacturing workloads receive priority even when energy optimization policies are active. This context-aware design is important because industrial sustainability cannot be achieved by indiscriminately reducing computational resources.

The conceptual results indicate that the proposed architecture can reduce normalized computational energy consumption, improve cluster utilization, decrease idle container activity, reduce cloud-bound IIoT data transmission, and shorten compute-active DevOps pipeline execution. The evaluation also suggests that critical predictive maintenance workloads can achieve improved responsiveness when scheduling decisions incorporate manufacturing priorities. These findings demonstrate that sustainable DevOps can simultaneously support environmental objectives and intelligent industrial operations when optimization is governed by service-level, safety, and production constraints.

The research further establishes connections among manufacturing process optimization, API engineering, digital twin systems, predictive maintenance, cloud-native orchestration, and green software engineering. The supplied foundational studies by Kumar, Gummadi, and Pokala provide important perspectives on machining optimization, API specification, digital twins, predictive maintenance, carbon-aware Kubernetes scheduling, sustainable GitOps, and energy management. By integrating these concepts into a unified architecture, the proposed framework contributes a broader systems perspective to sustainable smart manufacturing.

Future research can extend the framework through real-time carbon-intensity APIs, reinforcement learning-based workload scheduling, renewable energy forecasting, federated energy analytics, adaptive digital twin synchronization, and lifecycle carbon accounting. Large-scale industrial validation should also examine heterogeneous edge devices, multi-cloud clusters, private 5G networks, real manufacturing equipment, and geographically distributed production facilities. Additional research is required to develop standardized sustainability metrics for DevOps pipelines and industrial microservices. Nevertheless, the proposed architecture provides a comprehensive foundation for integrating energy awareness into the continuous development, deployment, monitoring, and optimization of next-generation Industrial IoT and smart manufacturing applications.

References

[1] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems," Advances in Computers, vol. 82, pp. 47–111, 2011.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[3] F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," IEEE Access, vol. 5, pp. 20418–20427, 2017.

[4] Narapareddy, V. S. R., & Yerramilli, S. K. (2021). SUSTAINABLE IT OPERATION SYSTEM. International Journal of Engineering Technology Research & Management (IJETRM), 5(10), 147- 155.

[5] H. K. Pokala, "Carbon-Aware Kubernetes Scheduling with Sustainable GitOps and Energy-Efficient DevOps for Green Cloud Computing Practices," Journal of Computational Analysis and Applications, vol. 29, no. 6, pp. 1497–1506, 2021.

[6] L. Bass, I. Weber, and L. Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley, 2015.

[7] J. Lee, H. Davari, J. Singh, and V. Pandhare, "Industrial Artificial Intelligence for Industry 4.0-Based Manufacturing Systems," Manufacturing Letters, vol. 18, pp. 20–23, 2018.

[8] Y. Lu, X. Xu, and L. Wang, "Smart Manufacturing Process and System Automation—A Critical Review," Annual Reviews in Control, vol. 47, pp. 221–237, 2019.

[9] Gummadi, V. P. K. (2020). API Design and Implementation: RAML and OpenAPI Specification. Journal of Electrical Systems, 16(4). https://doi.org/10.52783/jes.9329.

[10] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems," Manufacturing Letters, vol. 3, pp. 18–23, 2015.

[11] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," IEEE Access, vol. 8, pp. 108952–108971, 2020.

[12] Gummadi, V. P. K. (2020). API Design and Implementation: RAML and OpenAPI Specification. Journal of Electrical Systems, 16(4). https://doi.org/10.52783/jes.9329.

[13] Q. Qi and F. Tao, "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison," IEEE Access, vol. 6, pp. 3585–3593, 2018.

[14] Kumar, Anuj. "Digital Twin-Based Smart Manufacturing Systems for Real-Time Process Optimization." International Journal of Engineering Research and Development, vol. 10, no. 5, 2014, pp. 65–72.

[15] R. Verdecchia, P. Lago, C. Ebert, and C. de Vries, "Green IT and Green Software," IEEE Software, vol. 34, no. 6, pp. 16–18, 2017.

[16] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J. M. Pierson, and A. V. Vasilakos, "Cloud Computing: Survey on Energy Efficiency," ACM Computing Surveys, vol. 47, no. 2, pp. 1–36, 2015.

[17] R. Morabito, "Virtualization on Internet of Things Edge Devices with Container Technologies: A Performance Evaluation," IEEE Access, vol. 5, pp. 8835–8850, 2017.

[18] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-Scale Cluster Management at Google with Borg," EuroSys, pp. 1–17, 2015.

[19] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal Power Allocation in Server Farms," ACM SIGMETRICS, pp. 157–168, 2009.

[20] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual Machine Power Metering and Provisioning," Proceedings of the 1st ACM Symposium on Cloud Computing, pp. 39–50, 2010.