American Journal of Innovation in Science and Engineering (AJISE)

ISSN: 2158-7205 (ONLINE

VOLUME 5 ISSUE 3 (2026)

PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

American Journal of Innovation in Science and Engineering (AJISE)

Volume 5 Issue 3, Year 2026
ISSN: 2158-7205 (Online)
DOI: https://doi.org/10.54536/ajise.v5i3.8423
https://journals.e-palli.com/home/index.php/ajise

Automated Node Configuration via Predictive Analytics in Smart Grids: Separating Matched-Condition Accuracy from Operational Reliability

Svetlana Dmitrievna ORLOVA[1*]

Article Information

Received: June 12, 2026

Accepted: August 24, 2026

Published: September 02, 2026

Keywords

Automated Node Configuration, Autonomous Decision Making, Distribution State Estimation, Predictive Analytics, Smart Grid Reliability, Topology Generalization

ABSTRACT

Automated node configuration in smart grids, fault-triggered reconfiguration, tap-changer control, and switching decisions issued without operator confirmation, depends on predictive models validated primarily under structural correspondence between deployment and training topology. Reported precision for these models reaches 94.28% under matched topology and telemetry conditions. Under topology shift, F1-score for graph-based fault detectors decreases by 12% for attention-based architectures and by up to 60% for recurrent architectures without graph structure. A separate class of false data injection attacks, constructed against on-load tap-changer volt/var optimization, steers control decisions toward harmful setpoints without triggering residue-based bad-data detection. Two conditions determine operational reliability for automated node configuration: structural correspondence between deployment and training topology, and integrity of the measurements the model consumes. Reported evaluation practice combines both conditions into a single accuracy or precision figure. A validity-monitoring architecture, built on a distinction between aleatoric uncertainty and epistemic uncertainty, separates the two conditions into a topology-comparison check, evaluated by direct comparison of deployed and trained network structure, and an execution-time feasibility check computed independently of classifier confidence.

INTRODUCTION

A random forest classifier trained on load-redistribution data extracted from 180 randomized topologies parameterized by the IEEE 123 Node Test Feeder reaches 94.28% precision and 81.05% recall when evaluated on topology instances drawn from the same generative process used for training (Carrascal *et al.*, 2024). A recurrent-graph neural network trained on one configuration of the IEEE 123-bus system and tested on a distinct configuration loses up to 60% of its F1-score when the architecture reduces to a recurrent model without graph structure, and 12% to 25% when graph structure is preserved through attention-based message passing (Karabulut *et al.*, 2025). Fault-prediction models deployed to trigger switching or tap-changer adjustment inside a self-healing distribution feeder operate on network configurations selected by operating conditions. Automated node configuration depends on the integrity of the telemetry a predictive model consumes. A false data injection scheme constructed against on-load tap-changer-based volt/var optimization steers the optimization toward a harmful setpoint without triggering estimator-level bad-data detection (Choeum & Choi, 2019). A classification output produced from a given measurement stream carries the confidence score assigned by the classifier; that score is computed from the measurement stream supplied to the model and does not encode a separate check on whether the stream has been altered before reaching the model.

Mikhailiuk (2026) develops an architecture for prediction, coordination, and fault tolerance in distributed autonomous systems that positions decision latency, coordination stability, and degradation behavior as properties of a single design problem. Within that architecture, an explicit distinction between aleatoric uncertainty and epistemic uncertainty classifies noise in an otherwise correctly modeled process separately from a gap between what a model was trained to expect and what it currently observes. A framework for pre-positioned computation extends this distinction operationally: a system that prepares and caches a response to an anticipated event before the event occurs retains that response's validity only while current conditions remain within the range the response was prepared for (Mikhailiuk, 2026).

LITERATURE REVIEW

Fault-prediction and reconfiguration research for smart-grid node configuration is evaluated predominantly on topology instances sampled from the distribution used for training. Carrascal *et al.* (2024) generate 180 randomized network topologies constrained to the node counts, connectivity levels, and physical parameters of the IEEE 123 Node Test Feeder, train a Random Forest classifier with recursive feature elimination on load-redistribution data drawn from this family, and report 94.28% precision and 81.05% recall on held-out instances from the same generative process; the reported figure is an average across the sampled topology family. Nguyen *et al.* (2023) train temporal recurrent graph neural networks on the

[1] Head of the Digital Technologies Analytics Department, Institute of Applied Digital Research "Innovative Platforms", Russian Federation
[*] Corresponding author's e-mail: svetlana.d.orlova@gmail.com

Potsdam microgrid and the IEEE 123-node system and extract fault information from voltage measurements, a substitution that removes the requirement to instrument every line with a relay. Chanda and Soltani (2025) train a heterogeneous multi-task graph neural network to detect, locate, and classify faults on the IEEE-123 feeder and generate an explainability output identifying the nodes carrying the highest diagnostic weight. Gholizadeh and Musilek (2024) apply power-flow analysis to the fixed network structure to reduce a combinatorial switching action space before training a deep Q-network to select reconfiguration actions, producing a tractable training procedure at realistic bus counts.

Karabulut *et al.* (2025) construct a benchmark in which several recurrent-graph architectures, including GraphSAGE-, GAT-, and GATv2-based models, are trained on one topology configuration of the IEEE 123-bus system and tested on a distinct configuration, isolating topology shift as the sole varying factor between training and test conditions; the fault-detection task and dataset composition remain fixed across the two conditions. Jacob *et al.* (2024) train a capsule-based graph reinforcement learning controller for outage restoration and validate it on three structurally distinct networks, the 13-, 34-, and 123-bus modified IEEE systems, without retraining between the three structures. Aurangzeb *et al.* (2026) attach an uncertainty-quantification layer to graph-based fault detection, producing a confidence estimate alongside the point prediction under degraded and noisy operating conditions.

A second evaluation axis concerns the integrity of the measurements a predictive or estimation model consumes, independent of topology. Liang *et al.* (2017) define false data injection as a class of attack that manipulates sensor measurements to introduce undetected errors into state-variable estimates, constructing the injected error to remain consistent with the network's own physical model and evade residue-based bad-data detection. Zhao *et al.* (2018) extend the attack surface from linearized DC state-estimation models to nonlinear AC state estimators; detection methods calibrated against the linear case require modification to extend to the nonlinear case that distribution networks use. Deng *et al.* (2019) analyze distribution-level state estimation under realistic unbalanced-network conditions, the estimation layer automated node configuration in a distribution feeder depends on directly, and quantify the resulting constraints on detectability relative to balanced, transmission-level assumptions. Choeum and Choi (2019) construct a false data injection scheme aimed directly at on-load tap-changer-based volt/var optimization, the control decision, and demonstrate that the optimization can be steered toward a harmful setpoint without triggering the estimator-level detection mechanisms analyzed in the preceding three studies.

Self-healing distribution architectures constitute the operational context into which predictive and estimation outputs are consumed as inputs. Hamdeen *et al.* (2025) describe multi-agent self-healing frameworks organized around protection coordination and fault location, isolation, and service restoration, in which primary protection relies on peer-to-peer communication among zone agents that detect and isolate faults using locally available measurements. Arefifar *et al.* (2023) identify communication delay and completeness of system information as recurring practical constraints on self-healing performance. Li *et al.* (2021) decompose self-healing into phased sub-problems, detection, isolation, restoration, each addressed by a distinct algorithm within a unified multiagent framework.

Mikhailiuk (2026) develops an architecture for prediction, coordination, and fault tolerance in distributed autonomous systems, built around an explicit distinction between aleatoric uncertainty, noise in an otherwise correctly modeled process, and epistemic uncertainty, a gap between what a model was trained to expect and what it currently observes (Mikhailiuk, 2026). A Temporal Uncertainty Buffer implements this distinction operationally: a cached candidate policy remains valid while a measured deviation between the state a policy was computed for and the state currently observed stays under an explicit threshold, reported at approximately 0.2 to 0.25 in a continuous feature-space distance, with cache validity holding for 70% to 85% of decision cycles under stable operating conditions (Mikhailiuk, 2026). A hybrid architecture separates a neural component, producing a preference ranking over candidate actions, from a formally specified component that re-evaluates each candidate against fixed conditions at the moment of execution, constructed so the resulting safety guarantee does not depend on similarity between current and training conditions (Mikhailiuk, 2026). A Hierarchical Intent Graph filters candidate actions against structural and resource dependencies before coordination, eliminating 40% to 55% of candidate actions in simulated multi-agent deployments and reducing the number of actions entering coordination to three to eight per decision-making unit.

MATERIALS AND METHODS

Source selection considered peer-reviewed journal articles, peer-reviewed conference proceedings, and academic monographs addressing predictive fault detection, reconfiguration, or restoration in electrical distribution networks, false-data-injection or state-estimation vulnerability in power systems, and multi-agent self-healing distribution architectures, published between 2017 and 2026. Fifteen studies met these criteria: seven address predictive fault detection, reconfiguration, or restoration; four address false-data-injection or state-estimation vulnerability; three describe self-healing architectures that consume the outputs of the first group; and one describes a prediction-and-coordination architecture for autonomous multi-agent systems in transportation, manufacturing, and warehouse-logistics domains. Of the seven fault-detection, reconfiguration,

and restoration studies, three report a quantitative accuracy, precision, recall, or F1-score figure tied to an explicitly stated topology-test design; the remaining four report a methodological contribution, sensor substitution, multi-task diagnostic and explainability output, action-space pruning, or uncertainty quantification, without a topology-shift metric directly comparable to the other three.

Two evaluation axes structure the comparison: structural correspondence between the topology used to validate a model and the topology present at deployment, and integrity of the measurements a model consumes. Topology-test design for the three quantitatively comparable studies was classified into two categories: an averaged multi-instance design, in which a reported metric is computed across multiple topology instances drawn from one generative distribution, and a fixed-configuration design, in which a model is trained on one topology configuration and tested on a second, distinct configuration. For the four telemetry-integrity studies, the target layer of the reported attack or vulnerability, sensor measurement, state estimate, or control decision, was recorded directly from each study. Mechanisms described for the prediction-and-coordination architecture, the aleatoric/epistemic uncertainty distinction, execution-time feasibility checking, staged degradation classification, and hierarchical action-space filtering, were compared against the topology-test and telemetry-integrity categories established for the smart-grid-specific studies to identify which of these mechanisms a smart-grid predictive pipeline for automated node configuration implements.

RESULTS AND DISCUSSION

Table 1 reports the topology-test design and quantitative outcome for the three studies with directly comparable numeric results, together with the validity-monitoring parameters reported for a prediction-and-coordination architecture outside the smart-grid domain. Carrascal *et al.* (2024) report 94.28% precision and 81.05% recall, averaged across 180 topology instances drawn from one generative distribution. Karabulut *et al.* (2025) report an F1-score reduction of up to 60% for a recurrent architecture without graph structure and of 12% for a graph-attention architecture, each measured between one fixed training topology and a second, distinct test topology; the reductions correspond to structural mismatch between the training and test topology, since the fault-detection task and dataset composition remain fixed across the two conditions. Jacob *et al.* (2024) report a reduction in unserved energy of 607.45 kWs on a 13-bus system and 596.52 kWs on a 34-bus system relative to baseline restoration, measured without retraining across three distinct network structures. Mikhailiuk (2026) reports a cache validity of 70% to 85% of decision cycles for a Temporal Uncertainty Buffer discarded at a deviation threshold of 0.2 to 0.25 in feature-space distance, and a reduction of 40% to 55% in candidate actions entering coordination for a Hierarchical Intent Graph, each measured on transportation, manufacturing, and warehouse-logistics deployments, not on a topology-shift design.

Table 1: Topology-test design and reported quantitative result for predictive fault-detection, localization, and restoration models

Study	Model / method	Topology-test design	Reported result
Carrascal *et al.* (2024)	Random Forest, recursive feature elimination	180 topologies sampled from one generative distribution; metric averaged across the family	Precision 94.28%; recall 81.05%
Karabulut *et al.* (2025)	GRU (recurrent, no graph structure)	Trained on one IEEE 123-bus configuration, tested on a distinct configuration	F1-score reduction up to 60%
Karabulut *et al.* (2025)	GATv2 (graph attention)	Trained on one IEEE 123-bus configuration, tested on a distinct configuration	F1-score reduction of 12%
Jacob *et al.* (2024)	Capsule-based graph reinforcement learning controller	Validated on 13-, 34-, and 123-bus modified IEEE systems without retraining between structures	Unserved energy reduced by 607.45 kWs (13-bus) and 596.52 kWs (34-bus) relative to baseline restoration
Mikhailiuk (2026)	Hierarchical Intent Graph	Validated across transportation, manufacturing, and warehouse-logistics deployments	40%–55% of candidate actions eliminated before coordination; 3–8 actions per decision-making unit

Carrascal *et al.* (2024) design computes a metric averaged across topology instances drawn from one generative distribution. Karabulut *et al.* (2025) and Jacob *et al.* (2024) designs measure performance at one or more fixed, distinct topology configurations.

Table 2 reports the target layer and finding for the four

telemetry-integrity studies. Liang *et al.* (2017) report that an injected error constructed to match the network's physical model evades residue-based bad-data detection. Zhao *et al.* (2018) report that detection methods calibrated on linearized DC models require modification to extend to nonlinear AC estimators. Deng *et al.* (2019) report that detectability under realistic unbalanced-network conditions is constrained relative to balanced, transmission-level assumptions. Choeum and Choi (2019) report that an on-load tap-changer optimization can be steered to a harmful setpoint without triggering estimator-level detection mechanisms.

Table 2: Target layer and reported finding for false-data-injection and state-estimation studies relevant to automated node configuration.

Study	Attack / estimator class	Target layer	Reported finding
Liang *et al.* (2017)	False data injection, general taxonomy	Sensor measurements / state-variable estimates	An injected error constructed to match the network's physical model evades residue-based bad-data detection
Zhao *et al.* (2018)	Generalized false data injection	Nonlinear AC state estimator	Detection methods calibrated on linearized DC models do not transfer to nonlinear AC estimators without modification
Deng *et al.* (2019)	False data injection	Distribution-level state estimation, unbalanced network	Detectability under realistic unbalanced-network conditions is constrained relative to balanced, transmission-level assumptions
Choeum & Choi (2019)	OLTC-induced false data injection	Volt/var optimization (tap-changer control)	The optimization is steered to a harmful setpoint without triggering estimator-level detection

Telemetry integrity, summarized in Table 2, and topology correspondence, summarized in Table 1, are independent evaluation axes: an attack constructed against a state estimate or control decision operates on the measurement the model consumes and does not depend on whether the model's structural assumptions match the deployed topology. The self-healing frameworks in Hamdeen *et al.* (2025), Arefifar *et al.* (2023), and Li *et al.* (2021) consume the confidence values and state estimates produced by the fault-detection and estimation layers in Tables 1 and 2 as direct inputs to fault location, isolation, and restoration logic. The phase decomposition in Li *et al.* (2021) is organized around the sequence of the restoration process, detection, isolation, restoration.

Discussion

An accuracy figure averaged across topology instances drawn from one generative distribution measures expected performance over that distribution as a whole. Performance on an individual held-out instance, evaluated against a model trained on a distinct instance, falls below the distribution average whenever that instance sits in the lower part of the performance distribution; averaging combines these lower-performing instances with higher-performing instances into a single reported number. A design that trains on one fixed topology configuration and tests on a second, distinct configuration measures a single point in this distribution. The precision figure of 94.28% reported under an averaged multi-instance design (Carrascal *et al.*, 2024) and the F1-score reduction of up to 60% reported under a fixed-configuration design (Karabulut *et al.*, 2025) describe two different statistics of a performance distribution over topology instances: a mean value and a specific evaluated point.

The distinction between aleatoric and epistemic uncertainty (Mikhailiuk, 2026) classifies noise in an otherwise correctly modeled process separately from a gap between trained expectation and current observation. A Temporal Uncertainty Buffer, built on this distinction, caches a candidate policy while a measured deviation between the state a policy was computed for and the state currently observed remains under an explicit threshold, and discards the cached policy in favor of live computation once the threshold is exceeded. Applied to the fault-prediction models in Table 1, the corresponding mechanism is a monitored boundary around the network configuration for which a reported precision figure remains applicable; this boundary is a separate computational component from the classifier itself, absent from the evaluation designs in Carrascal *et al.* (2024), Nguyen *et al.* (2023), Chanda and Soltani (2025), and Gholizadeh and Musilek (2024).

A second mechanism separates a neural component, producing a preference ranking over candidate actions, from a formally specified component that evaluates each candidate against fixed conditions at the moment of execution (Mikhailiuk, 2026). The execution-time check applies to the current state at the moment of evaluation. A node-configuration decision generated by a classifier of the kind in Table 1 and passed through a protection-coordination feasibility check re-evaluated at execution time, the function IEC 61850-based fault-

location-isolation-and-restoration logic implements, requires a falsified measurement (Choeum & Choi, 2019) to produce a candidate action that also passes a physically grounded feasibility check computed independently of the classifier's confidence score.

The Temporal Uncertainty Buffer's validity threshold is calibrated on continuous feature-space distance, reported as a deviation of approximately 0.2 to 0.25 in that space, for domains in which structural change accumulates as continuous drift. Topology change in a power distribution network occurs at a discrete instant: a switching operation or a DER interconnection event changes the graph structure of the network at a single step. The topology-shift result reported by Karabulut *et al.* (2025) is defined as the difference in performance between one fixed topology and a second fixed topology. A single-edge topology change can leave a continuous embedding of network state only marginally displaced while invalidating the adjacency structure a graph neural network's learned parameters assume. A validity-monitoring primitive for power-distribution topology therefore requires direct comparison of the deployed topology's graph structure against the topologies represented in training, evaluated at the moment a candidate node-configuration decision is generated.

A three-tier degradation model classifies coordination between distributed decision-making units into a nominal regime, sustained while measured communication latency remains below 40 milliseconds and packet loss remains below 10%; a constrained regime, in which coordination is confined to locally connected subsets of units as packet loss rises to 10%-25% or latency rises to 40-80 milliseconds; and an isolated regime, in which each unit operates on local safety constraints and fallback policies once packet loss exceeds approximately 25% or communication disruptions persist beyond 150-200 milliseconds (Mikhailiuk, 2026). Fault location, isolation, and restoration logic built on peer-to-peer communication among zone agents depends on the connectivity of that communication channel for detecting and isolating faults; explicit latency and packet-loss thresholds of this kind specify the point at which zone-agent coordination is classified as unavailable instead of degraded, a quantified classification absent from the description of zone-agent coordination in Hamdeen *et al.* (2025).

A Hierarchical Intent Graph filters candidate actions against structural and resource dependencies before coordination takes place, eliminating 40% to 55% of candidate actions in simulated multi-agent deployments and reducing the number of actions entering coordination to three to eight per decision-making unit (Mikhailiuk, 2026). Power-flow analysis applied to a fixed network structure performs an equivalent filtering step for distribution-network reconfiguration, reducing a combinatorial switching action space to a tractable set before reinforcement-learning training (Gholizadeh & Musilek, 2024). Both filtering steps reduce the action space evaluated at decision time using a structural feasibility criterion computed from the network or task structure, independent of classifier confidence.

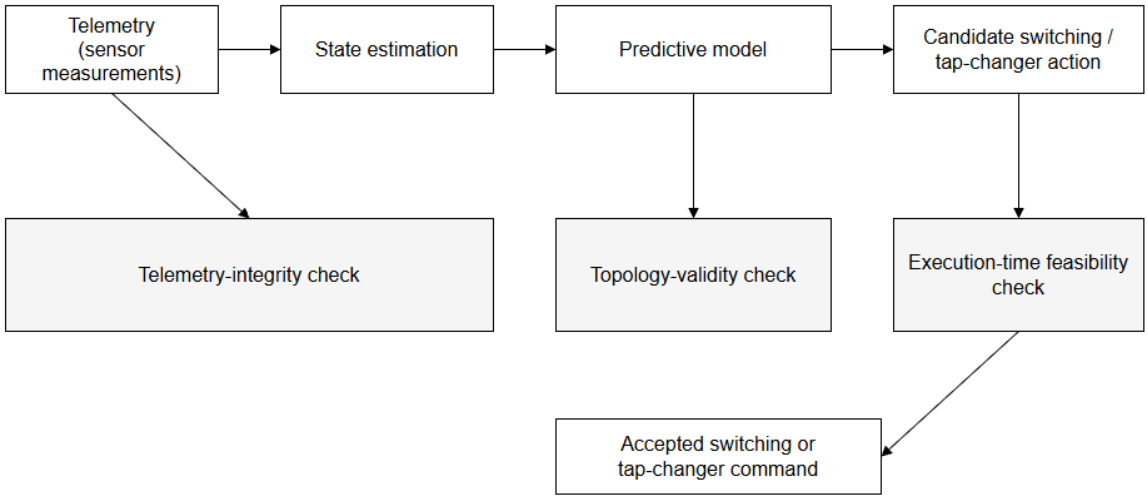

Figure 1: Validity-monitoring pipeline for automated node configuration: telemetry-integrity check at the state-estimation layer, topology-validity check at the predictive-model layer, and execution-time feasibility check at the point of action acceptance.

CONCLUSION

Automated node configuration is reliable under two conditions: structural correspondence between deployment and training topology, and integrity of the measurements the model consumes. A validity-monitoring architecture for automated node configuration combines a topology-comparison check, evaluating deployed graph structure against topologies represented in training before a predictive output is accepted as applicable, with an execution-time feasibility check, evaluating a candidate switching or tap-changer command against physical network constraints independently of classifier confidence. Figure 1 summarizes the resulting pipeline, positioning the telemetry-integrity check at the state-

estimation layer, the topology-validity check at the predictive-model layer, and the feasibility check at the point where a candidate action is accepted or rejected for execution.

Extending a train-on-one-configuration, test-on-another evaluation design from fault classification to reconfiguration and restoration models, including action-space-pruned reinforcement-learning agents and multiagent self-healing frameworks, on topology pairs constructed the same way, would quantify whether the 12%-to-60% degradation range measured for fault classification also applies to the switching and restoration decisions that act on that classification. A validity-monitoring architecture of the kind specified in Figure 1 gives that extension a direct engineering target: a topology-comparison check computed at the moment a candidate action is generated, and a feasibility check computed at the moment that action is accepted for execution.

REFERENCES

Amahrouch, A., Saadi, Y., & El Kafhali, S. (2025). Optimizing energy efficiency in cloud data centers: A reinforcement learning-based virtual machine placement strategy. *Network, 5*(2), Article 17. https://doi.org/10.3390/network5020017

Arroba, P., Moya, J. M., Ayala, J. L., & Buyya, R. (2017). Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers. *Concurrency and Computation: Practice and Experience, 29*(10), Article e4067. https://doi.org/10.1002/cpe.4067

Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience, 24*(13), 1397–1420. https://doi.org/10.1002/cpe.1867

Bodra, D., & Khairnar, S. (2025). Machine learning-based cloud resource allocation algorithms: A comprehensive comparative review. *Frontiers in Computer Science, 7*, Article 1678976. https://doi.org/10.3389/fcomp.2025.1678976

Carrión, C. (2022). Kubernetes scheduling: Taxonomy, ongoing issues and challenges. *ACM Computing Surveys, 55*(7), 1–37. https://doi.org/10.1145/3539606

Golec, M., Walia, G. K., Kumar, M., Cuadrado, F., Gill, S. S., & Uhlig, S. (2024). Cold start latency in serverless computing: A systematic review, taxonomy, and future directions. *ACM Computing Surveys, 57*(3), Article 65. https://doi.org/10.1145/3700875

Kahil, H., Sharma, S., Välisuo, P., & Elmusrati, M. (2025). Reinforcement learning for data center energy efficiency optimization: A systematic literature review and research roadmap. *Applied Energy, 389*, Article 125734. https://doi.org/10.1016/j.apenergy.2025.125734

Liu, X., Wen, J., Chen, Z., Li, D., Chen, J., Liu, Y., Wang, H., & Jin, X. (2023). FaaSLight: General application-level cold-start latency optimization for function-as-a-service in serverless computing. *ACM Transactions on Software Engineering and Methodology, 32*(5), Article 119. https://doi.org/10.1145/3585007

Mikhailiuk, M. (2026). *Applied artificial intelligence in modern infrastructure systems*. LAP LAMBERT Academic Publishing.

MirhoseiniNejad, S., Moazamigoodarzi, H., Badawy, G., & Down, D. G. (2020). Joint data center cooling and workload management: A thermal-aware approach. *Future Generation Computer Systems, 104*, 174–186. https://doi.org/10.1016/j.future.2019.10.040

Radovanović, A., Koningstein, R., Schneider, I., Chen, B., Duarte, A., Roy, B., Xiao, D., Haridasan, M., Hung, P., Care, N., Talukdar, S., Mullen, E., Smith, K., Cottman, M., & Cirne, W. (2023). Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems, 38*(2), 1270–1280. https://doi.org/10.1109/TPWRS.2022.3173250

Safari, A., Sorouri, H., Rahimi, A., & Oshnoei, A. (2025). A systematic review of energy efficiency metrics for optimizing cloud data center operations and management. *Electronics, 14*(11), Article 2214. https://doi.org/10.3390/electronics14112214

Senjab, K., Abbas, S., Ahmed, N., & Khan, A. U. R. (2023). A survey of Kubernetes scheduling algorithms. *Journal of Cloud Computing, 12*, Article 87. https://doi.org/10.1186/s13677-023-00471-1

Wan, J., Gui, X., Zhang, R., & Fu, L. (2018). Joint cooling and server control in data centers: A cross-layer framework for holistic energy minimization. *IEEE Systems Journal, 12*(3), 2461–2472. https://doi.org/10.1109/JSYST.2017.2700863