Review of Applied Science and Technology

***Volume:*2; *Issue:* 1**
***Pages:* 185– 226**

High-Performance Computing–Assisted Modeling and Real-Time Analysis of Electrical Power Networks and Industrial Control Systems

Md. Sultan Mahamud[1]; K M Tanvir Anjum Anick[2];

[1]. *Senior Electrical Engineer, Acumen Engineering Solution, Bangladesh.*
Email: sultan2961@gmail.com

[2]. *Operation Engineer & Trainer, Baraka Patenga Power Limited, Bangladesh*
Email: tanvir.anjum.anick1@gmail.com

Doi: 10.63125/727j5j39

Received: *15 January 2023;* ***Revised:*** *24 February 2023;* ***Accepted:*** *15 March 2023;* ***Published:*** *26 March 2023*

Abstract

This study examined the performance, robustness, and real-time feasibility of high-performance computing–assisted modeling and integrated analytical pipelines for electrical power networks coupled with industrial control systems. A quantitative experimental design was implemented using 360 structured run instances distributed across small, medium, and large benchmark grid tiers, multiple telemetry regimes, and standardized contingency libraries. Dependent outcomes included update latency, sustained throughput, state-estimation error, contingency ranking stability, and anomaly detection delay, while independent variables included compute architecture type, parallelism level, grid size tier, telemetry regime, contingency volume, and disturbance severity. Mixed-effects regression models with clustered random intercepts were used to quantify performance drivers. Results showed that mean update latency across the full sample was 148.2 ms (P95 = 268.0 ms), while sustained throughput averaged 6.8 updates per second. Hybrid CPU–GPU configurations reduced latency by 24.63 ms relative to CPU-only baselines ($p < .001$) and increased throughput by 1.18 updates per second. Large-grid tiers increased latency by 72.84 ms and reduced throughput by 4.27 updates per second compared with small-grid tiers ($p < .001$), confirming scale as the dominant workload driver. Convergence rates averaged 96.9%, indicating stable numerical robustness across tiers. Analytical quality improved under richer telemetry, with PMU-only configurations reducing estimation error by 0.008 per unit relative to SCADA-only ($p < .001$). Contingency ranking stability averaged 0.88, while missed critical contingencies averaged 0.37 per run under standardized runtime budgets. Anomaly detection delay averaged 312.5 ms and increased significantly under higher disturbance severity and larger grid tiers. Overall, the findings demonstrated that real-time cyber-physical feasibility in power–ICS environments was primarily governed by grid scale, telemetry design, and workload composition, while compute acceleration and parallelism provided statistically significant but workload-dependent performance improvements.

Keywords

High-Performance Computing, Power Networks, Industrial Control Systems, Real-Time Analytics, Cybersecurity.

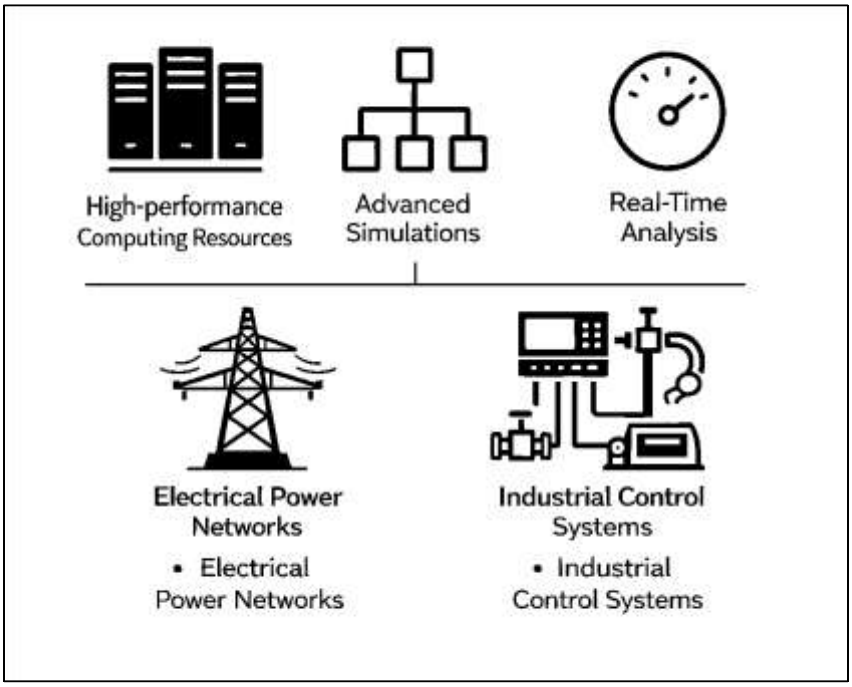
INTRODUCTION

High-performance computing (HPC) refers to the aggregation of advanced computational resources, including parallel processors, distributed memory architectures, and high-speed interconnect networks, to execute complex numerical operations at extremely high speeds. It is characterized by the ability to process large volumes of structured and unstructured data while solving multi-dimensional mathematical models that exceed the capabilities of conventional computing systems. In engineering domains, HPC is applied to simulations that require iterative matrix operations, nonlinear optimization, probabilistic modeling, and dynamic system integration (Huang et al., 2017). Electrical power networks are interconnected infrastructures composed of generation units, transmission lines, substations, transformers, and distribution feeders designed to deliver electrical energy from producers to consumers under strict stability and reliability constraints. Industrial control systems (ICS) constitute the cyber-physical layer responsible for supervising, regulating, and automating industrial and energy processes through programmable logic controllers, distributed control architectures, communication protocols, and sensor-actuator networks (Jin et al., 2017). When power networks and industrial control systems operate as integrated systems, they form large-scale, interdependent infrastructures where continuous electrical dynamics interact with discrete control commands and communication signals. Modeling such systems requires mathematical representations of power flow equations, state-space dynamic models, protection coordination schemes, and control logic structures. As grid size and automation depth increase, computational demands escalate due to the need to simulate thousands of buses, generators, loads, and real-time telemetry signals simultaneously (Mojlish et al., 2017). HPC-assisted modeling therefore represents a methodological integration in which scalable computational resources are employed to solve high-dimensional system equations and execute large-scale simulations within operationally relevant time constraints. Real-time analysis within this context involves continuous system monitoring, rapid numerical evaluation, and immediate processing of operational contingencies, enabling computational alignment with physical system dynamics (Zhou et al., 2019).

Electrical power networks and industrial control systems hold global strategic significance because they constitute the operational backbone of modern economies, public services, manufacturing sectors, and digital infrastructures. National development indicators, including industrial productivity, healthcare service delivery, transportation efficiency, and telecommunications stability, are directly dependent on reliable electricity supply and coordinated control architectures. In both developed and emerging economies, energy infrastructure expansion has intensified the scale and complexity of grid systems, integrating renewable energy plants, cross-border interconnections, and high-voltage transmission corridors (Yang & Chien, 2016). The globalization of energy markets has increased the need for accurate modeling of interconnected regional grids, where disturbances in one geographical area can propagate across national boundaries through synchronized transmission systems. Industrial control systems extend beyond power plants and substations into oil refineries, water treatment facilities, chemical manufacturing plants, and large-scale production environments. The reliability of these control architectures determines operational safety, process efficiency, and economic continuity. As infrastructure digitization accelerates, the volume of telemetry data generated by sensors, smart meters, phasor measurement units, and automated controllers has expanded significantly (Zhou et al., 2019). This growth in real-time data streams increases computational requirements for monitoring, analytics, and fault detection. International regulatory frameworks emphasize grid stability, cybersecurity resilience, and energy efficiency, reinforcing the need for computational tools capable of supporting compliance monitoring and operational auditing. The global integration of renewable generation sources further increases variability in system behavior, requiring dynamic analysis of voltage stability, frequency regulation, and load balancing (Pratt et al., 2016). HPC-assisted modeling becomes internationally significant because it provides the computational capacity necessary to simulate interconnected infrastructures at continental scales while maintaining analytical precision. The ability to process large datasets and execute synchronized simulations across distributed computing clusters supports cross-border coordination, system reliability assessment, and comprehensive risk evaluation in globally interconnected energy ecosystems (Monti et al., 2018).

Electrical power network modeling is grounded in mathematical formulations that describe steady-state and dynamic behavior of interconnected electrical components. The steady-state behavior of transmission networks is commonly represented through nonlinear algebraic equations derived from Kirchhoff's laws, admittance matrices, and complex power balance constraints. Solving these equations for large-scale systems requires iterative numerical techniques that involve repeated matrix factorizations and convergence evaluation across thousands of nodes (Wang et al., 2015).

Figure 1: High-Performance Computing in Power Systems



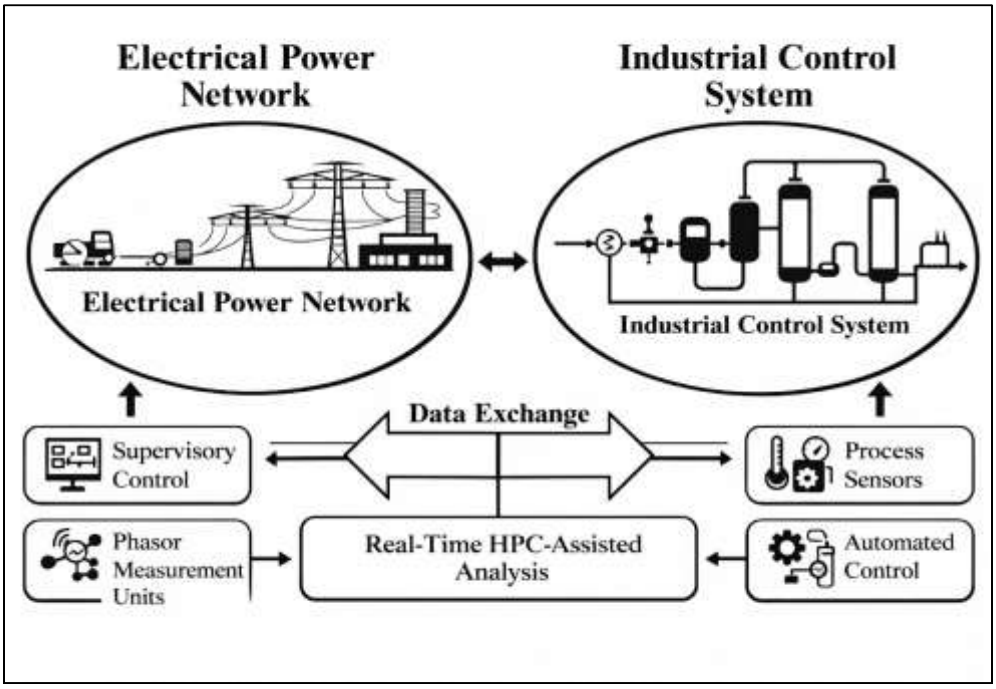
Dynamic modeling extends this representation by incorporating differential equations that describe generator rotor dynamics, excitation systems, turbine governors, and load frequency control mechanisms. Transient stability analysis requires time-domain simulations with small integration step sizes to capture oscillatory responses following disturbances such as faults or sudden load changes. Industrial control systems introduce additional computational complexity through discrete event logic, communication latency modeling, and feedback control loops. Supervisory control layers must process sensor measurements, execute control algorithms, and dispatch actuation signals within milliseconds (Czarnul et al., 2019). When combined with physical grid equations, these control processes form hybrid systems containing continuous dynamics and discrete transitions. HPC enables parallel decomposition of network matrices, distributed solution of state estimation problems, and concurrent execution of contingency scenarios. Domain decomposition methods partition large grids into computational subregions processed simultaneously across cluster nodes. Sparse matrix optimization and high-speed memory access reduce computational overhead in large admittance matrix calculations. GPU acceleration enhances the speed of repetitive numerical operations such as Jacobian evaluations and time-step integration (Sayed & Gabbar, 2017). These computational strategies allow large-scale network simulations to be completed within timeframes compatible with operational decision cycles. As network size increases, computational scaling becomes a central design consideration, making HPC-assisted frameworks essential for maintaining modeling fidelity without compromising execution speed (Poudel et al., 2017).

Real-time analysis in electrical power networks and industrial control systems refers to the continuous computational processing of live operational data within strict temporal constraints aligned with physical system behavior. Power systems operate at fixed frequencies, requiring constant balancing of generation and load to prevent frequency deviations and voltage instability. Real-time state estimation integrates measurements from supervisory control systems and phasor measurement units to reconstruct system operating conditions at high refresh rates. Industrial control systems simultaneously monitor process variables such as temperature, pressure, current, and voltage, executing automated adjustments to maintain predefined operational setpoints (Faysal & Shamsunnahar, 2022; Habibullah & Zaheda, 2022; Sun et al., 2020). The synchronization of computational analysis with physical system dynamics demands deterministic processing speed and minimal latency. HPC environments support this requirement by distributing computational tasks across multiple processors, enabling simultaneous evaluation of contingency scenarios, load flow recalculations, and dynamic security assessments. Real-time contingency analysis evaluates potential line outages or equipment failures before they occur, using parallel simulations to assess stability margins (Jahangir & Md Shahab, 2022; Ratul, 2022; Sun et al., 2020). Data assimilation techniques combine measurement streams with predictive models to update system states iteratively. High-throughput data pipelines ensure that sensor data is processed without bottlenecks, while optimized scheduling algorithms allocate computational resources dynamically. In industrial plants, real-time anomaly detection relies on pattern recognition algorithms operating on high-dimensional sensor arrays. HPC clusters provide the memory bandwidth and processing power necessary to execute such algorithms continuously. The convergence of physical processes and computational cycles requires coordinated timing mechanisms and robust communication architectures (Palmintier et al., 2016). Real-time HPC-assisted analysis therefore represents the operational integration of computational scalability with grid physics and automated control systems, forming a synchronized analytical framework capable of responding to disturbances within milliseconds.

Electrical power networks integrated with industrial control systems constitute complex cyber-physical systems characterized by layered interdependencies between hardware components, software logic, and communication networks. Physical components such as generators and transformers exhibit electromechanical dynamics governed by physical laws, while control layers interpret measurements and implement algorithmic decisions (Rossi et al., 2019). Communication networks transmit control signals and telemetry across geographically distributed infrastructures. Modeling this integrated environment requires simultaneous representation of electrical equations, control algorithms, and communication protocols. System complexity increases with the introduction of distributed energy resources, smart grid technologies, and automated substations. Each added node contributes to combinatorial growth in potential interaction states and contingency pathways. The dimensionality of system state vectors expands as additional sensors and controllers are deployed. HPC-assisted modeling addresses this complexity by enabling multi-layer simulation environments where electrical, control, and communication models are solved concurrently (Ratul & Subrato, 2022; Tahmina Akter Bhuya & Rebeka, 2022; Zhou et al., 2020). Parallel co-simulation frameworks allow distinct subsystems to be executed on separate computational nodes while exchanging synchronization signals. Such integration supports holistic evaluation of cascading failures, control misconfigurations, and load imbalance propagation. Large-scale scenario generation techniques explore thousands of operational configurations to evaluate stability boundaries. Industrial automation environments further introduce cybersecurity considerations, requiring computational modeling of intrusion detection algorithms and network resilience assessments. The computational burden associated with evaluating these interconnected domains in a unified framework exceeds traditional processing capabilities (F. Yang et al., 2019). HPC infrastructures provide scalable processing clusters capable of executing high-fidelity integrated simulations, ensuring that cyber-physical interactions are analyzed comprehensively within operationally relevant time constraints.

Quantitative research in HPC-assisted power network modeling relies on mathematically rigorous frameworks designed to produce reproducible and numerically stable results. Deterministic models solve algebraic and differential equations representing power flow and machine dynamics, while stochastic models incorporate uncertainty in load demand, renewable generation variability, and equipment reliability (Abir et al., 2021). Monte Carlo simulation techniques require repeated sampling and iterative computation, increasing computational demand exponentially as system size grows. Optimization-based frameworks such as optimal power flow involve nonlinear objective functions constrained by equality and inequality conditions. Solving these formulations requires iterative algorithms with convergence checks and sensitivity analysis. Numerical precision is critical because rounding errors in large sparse matrices can propagate and influence stability margins. HPC platforms support high-precision arithmetic and distributed memory architectures that manage large datasets without performance degradation. Parallel optimization solvers divide constraint evaluations among processors, accelerating convergence (Samanta et al., 2021). Time-domain simulations with adaptive step sizing enhance accuracy in transient stability analysis. Industrial control systems contribute additional quantitative layers through proportional-integral-derivative control equations, discrete logic transitions, and event-triggered responses. Integrating these models into a unified computational framework requires synchronization algorithms and numerical consistency checks. Validation of model outputs involves statistical comparison between simulated and observed operational data. HPC-assisted quantitative modeling ensures that large datasets can be processed with minimal computational delay while preserving numerical integrity (Cai & Braun, 2019). This computational reliability forms the methodological foundation for rigorous quantitative evaluation of system performance metrics such as voltage deviation, frequency stability, response latency, and fault recovery duration.

Figure 2: HPC-Integrated Cyber-Physical Power Framework



Infrastructure reliability in electrical power networks and industrial control systems is assessed through quantitative performance metrics that measure stability, availability, resilience, and operational efficiency. Reliability indices evaluate frequency and duration of service interruptions, while stability metrics assess voltage regulation and rotor angle coherence during disturbances. Industrial control environments measure response time, control accuracy, and process variability (Abir et al., 2021). Large-scale infrastructures generate extensive datasets reflecting operational states across thousands of nodes and sensors. Evaluating system-wide performance requires aggregating and analyzing these datasets using high-speed computational platforms. HPC-assisted environments enable parallel computation of reliability indices across multiple contingency scenarios. Sensitivity analysis quantifies how parameter variations influence system stability margins. Real-time performance dashboards integrate computational outputs into operator interfaces, facilitating continuous monitoring. Performance benchmarking involves repeated simulation of fault events under varying load and generation conditions (Samanta et al., 2021). Computational scalability ensures that expanded system models can be evaluated without compromising processing speed. Power networks interconnected across regions require coordinated reliability assessment to prevent cascading failures. Industrial plants dependent on uninterrupted power supply rely on precise synchronization between electrical infrastructure and control logic. Quantitative assessment of these integrated systems demands computational environments capable of handling large matrices, high-frequency data streams, and multi-scenario simulations simultaneously. HPC-assisted modeling therefore serves as the computational backbone for evaluating reliability, stability, and operational performance within complex, globally significant electrical and industrial infrastructures (Cai & Braun, 2019).

The objective of this quantitative research is to develop, operationalize, and evaluate a high-performance computing–assisted modeling and real-time analytical framework that can represent the coupled behavior of electrical power networks and industrial control systems with measurable computational efficiency, numerical reliability, and analytical accuracy. A primary objective is to formalize a unified mathematical representation that integrates steady-state power flow constraints, dynamic stability equations, and discrete industrial control logic into a single computational workflow that supports synchronized execution across large network topologies. A second objective is to quantify computational scalability by measuring execution time, speedup, memory utilization, and parallel efficiency under increasing system size, increasing data rates, and increasing scenario volume, using controlled benchmarking designs that enable repeatable performance comparisons. A third objective is to operationalize real-time analytics by defining strict latency and throughput targets and then empirically testing whether state reconstruction, contingency screening, and anomaly scoring can be executed within time budgets consistent with operational telemetry refresh cycles. A fourth objective is to evaluate numerical robustness by measuring convergence rates, residual errors, stability indicator consistency, and sensitivity to numerical precision settings when solving nonlinear algebraic and differential equations under stressed operating conditions. A fifth objective is to quantify the analytical value of integrated cyber-physical modeling by comparing results from isolated electrical-only simulations and coupled power-and-control simulations using standardized outcome measures such as voltage deviation distributions, frequency excursion magnitudes, control response delays, and disturbance recovery times. A sixth objective is to define a reproducible dataset construction and preprocessing pipeline for streaming measurements, including timestamp alignment, missing-data handling, and noise characterization, and then test how these preprocessing choices affect estimation accuracy and detection performance. A seventh objective is to establish a metrics-driven evaluation protocol that links HPC configuration parameters and model design choices to observable quantitative outcomes, enabling statistical testing of performance differences across configurations, workloads, and disturbance classes. Collectively, these objectives structure the study around measurable variables, testable computational and system-behavior hypotheses, and empirically comparable outcomes that align with quantitative research requirements for integrated modeling and real-time analysis of power networks and industrial control systems.

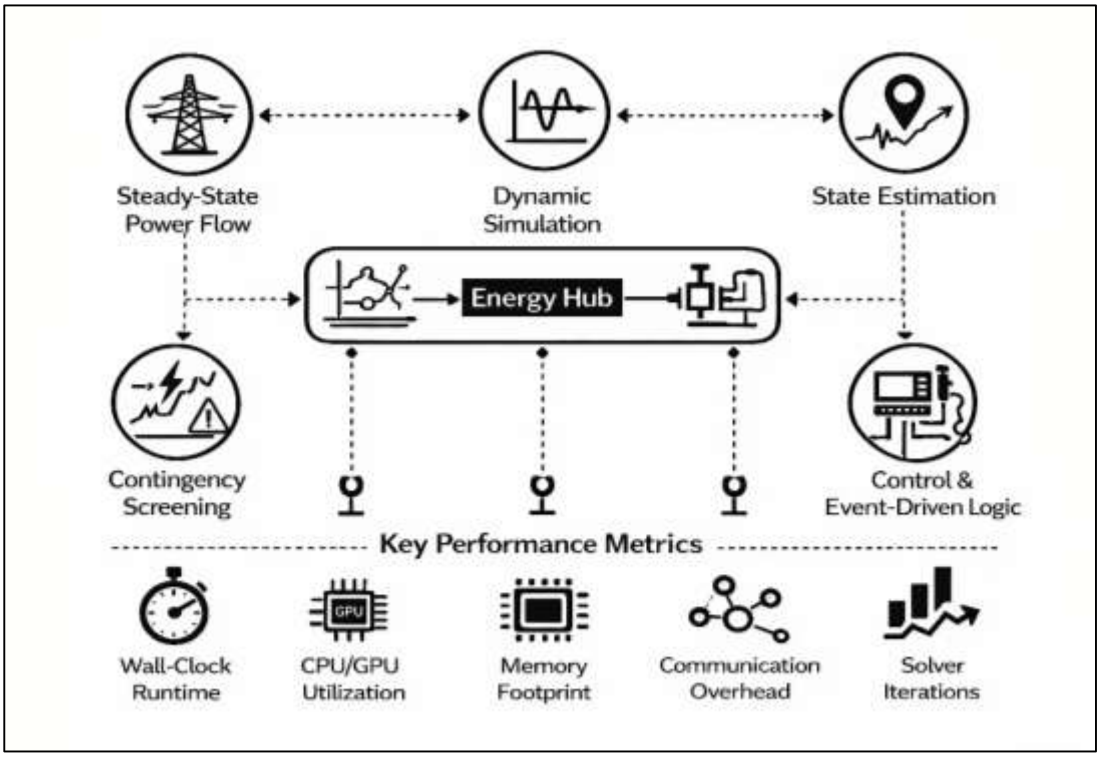
LITERATURE REVIEW

This Literature Review section synthesizes quantitative research that underpins high-performance computing–assisted modeling and real-time analysis of electrical power networks integrated with industrial control systems. The section is organized to align computational methods with measurable outcomes, emphasizing operational variables such as runtime, latency, throughput, numerical error, convergence behavior, detection accuracy, scalability, and robustness under disturbance scenarios (Ma et al., 2015). Rather than describing technologies in general terms, the review is structured around how prior studies have operationalized models, datasets, and evaluation protocols, including benchmark system sizes, sampling rates, contingency sets, and performance metrics. Each subsection is framed to support a quantitative paper by identifying measurable constructs, typical dependent and independent variables, and the dominant empirical designs used in the literature (e.g., simulation-based experiments, comparative benchmarking, sensitivity analysis, Monte Carlo evaluation, and real-time streaming tests) (Krama et al., 2018). The result is an evidence-guided structure that connects power-system physics and control logic with parallel computing architectures and real-time analytics pipelines, while keeping the focus on how researchers measure performance, validate models, and report statistical or computational significance.

Computational Workloads in Power-ICS Co-Simulation

The literature on high-performance computing applications in power system and industrial control system co-simulation consistently categorizes computational workloads into distinct but interrelated classes reflecting physical modeling, control logic execution, and real-time analytics demands. Steady-state power flow computation remains the foundational workload, representing algebraic resolution of network operating conditions under balanced assumptions (Wan et al., 2014).

Figure 3: HPC Workloads and Metrics for Power and Control Systems



Dynamic transient simulation expands this workload to time-domain modeling of generator electromechanical behavior and fault responses. State estimation workloads incorporate telemetry assimilation and residual evaluation under measurement uncertainty. Contingency screening tasks evaluate predefined outage sets to assess stability margins and security indices. Control-loop execution workloads simulate programmable logic controller scan cycles and supervisory control decisions. Event-driven ICS logic introduces discrete-event scheduling, alarm triggering, and communication message processing into the co-simulation environment (Kuruvila et al., 2021). Prior research shows that combining these workload classes in integrated cyber-physical simulation substantially increases computational intensity because continuous electrical dynamics and discrete control transitions must be synchronized within unified execution cycles. Studies examining co-simulation frameworks emphasize that workload heterogeneity leads to variable computational density, with dynamic simulations consuming greater processing resources per simulated second compared to steady-state power flow tasks. Parallel computing research in power engineering demonstrates that workload decomposition strategies are essential for maintaining computational tractability as system size expands. Empirical benchmarking in integrated environments further shows that combining contingency screening with dynamic response modeling increases execution complexity nonlinearly, particularly when large disturbance sets are evaluated sequentially (Liu et al., 2016). Research in distributed simulation architecture identifies workload classification as a prerequisite for effective task scheduling and resource allocation. These classifications provide a measurable structure for profiling execution time, synchronization overhead, and resource utilization across integrated power–ICS computational studies.

Quantitative investigations consistently identify structural and operational parameters as primary drivers of computational workload magnitude in power–ICS co-simulation. Network size indicators such as bus count, generator count, branch density, and protection device inclusion directly affect matrix dimensionality and solver iteration requirements (Mikkili et al., 2015). Industrial control integration introduces additional scaling factors including the number of control input/output points, PLC logic blocks, communication nodes, and event triggers. Telemetry sampling rate significantly influences state estimation complexity, as higher refresh intervals require more frequent matrix updates and residual recalculations. Contingency set size expands evaluation loops in security analysis studies, increasing execution time in proportion to the number of predefined outage scenarios. Simulation horizon length and time-step granularity determine the total number of integration cycles in transient analysis tasks. Empirical HPC benchmarking studies demonstrate that small reductions in time-step length lead to exponential growth in total computational iterations when long disturbance windows are simulated (Xiao et al., 2020). Solver iteration limits and convergence tolerances further influence computational load, with tighter tolerance thresholds increasing iteration counts per solution cycle. Research comparing weak and strong scaling conditions reveals that workload drivers interact multiplicatively, particularly when large contingency libraries are combined with high-frequency telemetry assimilation. Studies examining synthetic large-scale test systems indicate that bus count and contingency volume together form dominant predictors of runtime variability. Control system complexity also contributes significantly to synchronization overhead when discrete events are interleaved with continuous numerical solvers (Stellato et al., 2016). Across quantitative investigations, workload drivers are measured systematically to determine scalability boundaries, hardware efficiency thresholds, and performance bottlenecks within HPC-assisted co-simulation environments.

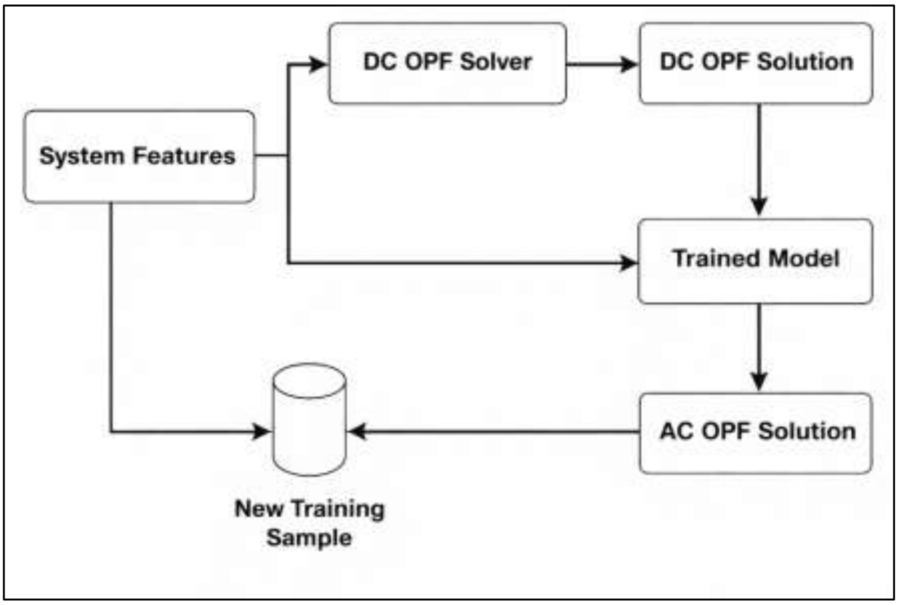
The literature consistently reports standardized performance metrics to evaluate HPC-assisted power–ICS modeling efficiency. Wall-clock runtime remains the primary metric, enabling comparison of execution duration across hardware configurations and algorithmic strategies. Studies frequently disaggregate runtime into computation, communication, and synchronization components to isolate performance bottlenecks. CPU and GPU utilization rates are commonly measured to assess hardware resource saturation and parallel efficiency (Lucia et al., 2020). Memory footprint is evaluated to determine scalability feasibility when simulating high-dimensional state vectors and contingency batches. Network communication overhead is quantified in distributed cluster environments to evaluate data transfer latency between computational nodes. Solver iteration counts and convergence cycles are reported to characterize numerical stability and computational effort per solution stage. Empirical comparisons across HPC architectures demonstrate that parallel efficiency declines when communication overhead exceeds computation time, particularly in tightly coupled co-simulation frameworks. Benchmarking studies also report throughput metrics such as contingencies processed per second and state estimation updates per unit time (Stellato et al., 2016). Latency profiling is used in real-time analysis research to evaluate compliance with operational refresh intervals. Statistical variance of runtime across repeated trials is often included to demonstrate reproducibility and deterministic performance. Comparative evaluations between CPU-only and hybrid CPU–GPU systems reveal differences in iteration stability and execution scaling. These quantitative measurement practices establish a consistent empirical basis for comparing modeling frameworks, hardware configurations, and solver optimizations in integrated power and control system simulations (De Carne et al., 2019).

Quantitative research in HPC-assisted co-simulation emphasizes the construction of standardized workload profiles to ensure comparability and reproducibility. Many studies adopt predefined contingency libraries, often based on N-1 or expanded disturbance sets, to maintain consistency across experimental trials. Fixed phasor measurement unit sampling rates are frequently used in state estimation benchmarking to stabilize data ingestion volume. Disturbance templates such as three-phase faults, line trips, and generator outages are scripted with identical timing parameters to allow direct runtime and stability comparisons (Ceseña & Mancarella, 2018). Synthetic large-scale grid models are commonly used to provide controlled scaling experiments while avoiding confidentiality constraints associated with operational systems. Research in distributed simulation frequently employs identical solver configurations and tolerance thresholds across hardware platforms to isolate architectural performance differences. Studies evaluating co-simulation synchronization often define standardized communication intervals between electrical and control layers to minimize variability. Benchmark frameworks sometimes include predefined simulation horizons to ensure equal computational exposure across methods (Alizadeh et al., 2020). Comparative analyses show that consistent workload construction reduces experimental noise and improves statistical reliability when assessing performance improvements. Repeated trials under identical disturbance sets allow calculation of runtime variance and hardware efficiency metrics. Standardized workload profiling therefore functions as a methodological foundation in HPC power–ICS research, enabling structured comparison of execution time, scalability limits, and resource utilization patterns across diverse computational architectures and modeling strategies (Kouro et al., 2015).

Parallel Power-Flow and Optimal Power Flow (OPF)

Quantitative literature on parallel AC power flow computation focuses on how to restructure the nonlinear network solution process so that large systems can be solved faster without sacrificing numerical reliability (L. Liu et al., 2020).

Figure 4: Parallel Power Flow and OPF Framework



Research commonly distinguishes between approaches that parallelize within a single solve (for example, parallelizing the linear algebra inside Newton-type iterations) and approaches that parallelize across many independent solves (for example, solving many operating points or contingencies concurrently). Within-solve methods often rely on decomposition strategies that partition the network or the numerical workload, such as bus or branch partitioning, domain decomposition, and multi-area coordination, so that different compute resources work on different parts of the problem at the same time. Another major line of work treats sparsity as the central performance lever, emphasizing that the admittance and Jacobian matrices in power flow are sparse and structured, enabling specialized sparse factorization and ordering techniques that reduce fill-in and accelerate repeated solves (Sulligoi et al., 2016). Distributed sparse linear algebra appears as a recurring methodological backbone, particularly for transmission-scale networks where memory footprint and factorization time can dominate runtime. Quantitative studies also report that parallel performance depends strongly on communication patterns between computational partitions: loosely coupled partitions can scale better, while tightly coupled iterative exchanges can create overhead that limits gains. Across this literature, the central contribution is not merely faster computation, but measurable reductions in runtime under controlled test cases, with detailed profiling that isolates where time is spent across factorization, updates, and convergence checks (Zhang et al., 2015).

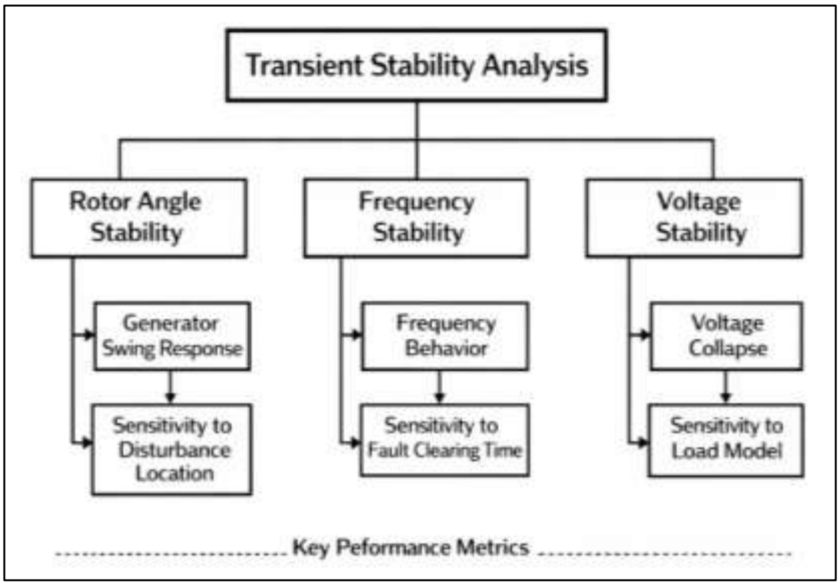
Parallel optimal power flow research extends beyond solving the network equations to managing constrained optimization workloads that include generator limits, voltage bounds, thermal line ratings, and security requirements. Quantitative studies commonly compare optimization formulations and solvers according to how constraints are evaluated, enforced, and distributed across compute resources (Jo et al., 2015). One cluster of research emphasizes decomposition-based OPF, where the system is divided into regions or subproblems coordinated by algorithmic mechanisms that reconcile boundary variables and shared constraints. Another cluster emphasizes parallel constraint screening and evaluation, especially for formulations where the constraint set becomes very large under security constraints or contingency modeling. In these settings, the computational bottleneck often shifts from the core optimization step to the repeated evaluation of feasibility and the repeated linear algebra associated with constraint handling (Huppmann et al., 2019). The literature also treats solver stability as a measurable outcome, tracking whether optimization runs terminate reliably under stressed operating points, ill-conditioned cases, or tight operational limits. Quantitative reporting often includes how frequently constraints become active, how constraint violations behave during iterations, and how sensitive outcomes are to initialization and tolerance settings. Overall, this body of work frames parallel OPF as an engineering problem in scalable numerical optimization, where improvements are validated through repeatable benchmarks, solver diagnostics, and careful measurement of computational costs tied to constraint-handling operations (Karamanakos et al., 2014).

Empirical evaluation in the parallel power flow and OPF literature is typically structured around measurable computational outcomes that capture both performance and numerical quality. Runtime reduction is treated as the primary performance signal, but studies often go further by reporting how performance changes as the number of processors, nodes, or accelerators increases, revealing practical scalability limits (Hu et al., 2015). Many investigations show that strong-scaling performance tends to saturate when communication, synchronization, or memory bandwidth becomes comparable to the useful compute time, while weak-scaling tests focus on whether runtime remains stable as problem size grows with compute resources. In addition to performance metrics, quantitative studies routinely track numerical behavior such as convergence rate patterns, iteration counts, and solver robustness across diverse operating conditions. For OPF specifically, optimization quality is assessed using measurable indicators like objective value consistency across runs, gap-like measures reflecting distance from best-known solutions, and feasibility properties such as constraint satisfaction and stability of constraint enforcement (Abadal et al., 2016). Solver stability is treated as an empirical property: the fraction of cases that converge, sensitivity to starting points, and behavior under stressed conditions all appear as measurable criteria. This dual emphasis—performance plus numerical integrity—reflects a consistent theme in the literature: parallelization must be justified not only by faster runtime but also by stable convergence and reliable constraint satisfaction across benchmark suites (Ristov et al., 2016).

Time-Domain Transient Stability on HPC

Time-domain transient stability analysis is treated in the literature as a computationally intensive class of dynamic security assessment because it requires tracking fast electromechanical trajectories immediately before, during, and after system disturbances. Studies commonly structure these simulations around fault-on and fault-cleared trajectories, where network topology changes temporarily and then returns to a post-fault configuration, forcing repeated re-evaluation of system states under shifting algebraic constraints (McLaughlin & Bader, 2014). The generator swing response and frequency behavior are central targets because they reflect the interaction between machine inertia, control action, and network coupling, while voltage collapse indicators are examined through the evolution of bus voltages and reactive power balance during stressed operating conditions. The computational intensity increases because these simulations do not produce a single operating point; they produce high-resolution trajectories over a disturbance window that must be captured with careful numerical integration and repeated network solution steps (Simic et al., 2019).

Figure 5: HPC-Based Transient Stability Analysis



Research in power-system dynamics frames this workload as dominated by iterative updates of dynamic component states paired with repeated network equation solutions, where sparse linear algebra and factorization steps can become the primary runtime contributor as system size grows. HPC-focused studies interpret transient stability as a natural target for parallelism because each simulation step contains structured linear algebra and because many contingency cases can be evaluated in batches under dynamic security assessment workflows (Economon et al., 2016). GPU- and cluster-based studies further describe the transient stability workload as a blend of compute-heavy per-step operations and communication-heavy synchronization when distributed partitions or heterogeneous CPU–GPU pipelines are used. Across this body of work, transient stability on HPC is portrayed not as a single algorithmic problem but as a pipeline combining disturbance scripting, dynamic model evaluation, network solution, event handling, and output extraction, all of which contribute measurable runtime and numerical sensitivity under large-scale conditions (Cassidy et al., 2014).

Quantitative transient stability studies consistently emphasize that time-step choice is a dominant factor shaping numerical accuracy, stability, and reproducibility of simulation trajectories. Smaller time steps typically reduce local approximation error and improve the fidelity of fast oscillatory dynamics, while larger time steps may mask short time-constant behavior and alter the apparent damping or frequency content of the response (Salza & Ferrucci, 2019). The literature frames this as a measurement problem: integration error is assessed using trajectory deviation, state residual behavior, and stability indicator consistency across repeated runs under different step sizes. Step-size stability is also treated as an empirical outcome, evaluated through whether the numerical method remains bounded and whether it converges reliably under stiff dynamic models that contain a wide spread of time constants. In operationally oriented studies, runtime per simulated second becomes a critical metric because it links numerical choices to feasibility under real-time or near-real-time requirements; time-step reductions increase the number of updates and network solves, often expanding runtime substantially for long disturbance windows (Cano, 2018). Event detection latency is addressed as a practical numerical outcome in studies where discrete events such as fault application, fault clearing, controller limit activation, or protection actions must be triggered at precise times; coarse time steps can delay or smear event timing and change post-event trajectories. Trajectory divergence thresholds are frequently used to describe when a simulation becomes untrustworthy, such as when numerical oscillations, drift, or discontinuity artifacts appear under aggressive step sizing or poorly conditioned network states. Across the literature, these outcomes form a consistent measurement set: error behavior, boundedness, timing accuracy of events, and runtime cost, all of which are used to justify the numerical integration configuration for transient stability workloads (Liu & Wang, 2015).

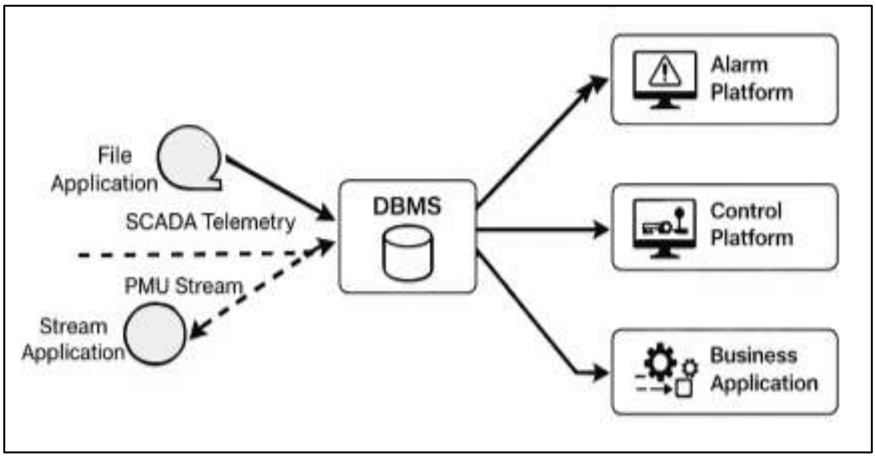
Sensitivity studies are a recurring design pattern in transient stability research because the dynamic response of a power system is highly dependent on where and how disturbances occur and on how component models represent physical behavior. Disturbance location studies evaluate how faults at different buses or along different corridors produce different oscillatory modes, separation risks, and voltage recovery profiles, making location a measurable driver of stability margins and simulation runtime variability (Liu & Wang, 2015). Fault clearing time sensitivity is treated as particularly influential because slight changes in clearing duration can shift systems from stable recovery to loss of synchronism, while also changing how long the simulation remains in a topology-altered configuration. Load model selection is repeatedly highlighted as a major source of uncertainty because static load assumptions, voltage-dependent loads, and composite dynamic load representations can yield materially different trajectories for frequency response and voltage recovery, affecting both physical realism and numerical stiffness (Chen et al., 2014). Parameter perturbation experiments examine how variations in generator inertia, damping, excitation response, governor settings, and control limits modify transient outcomes, often using structured perturbation ranges to quantify response spread and identify highly sensitive parameters. In HPC and benchmarking-oriented studies, these sensitivity protocols also serve a second purpose: they generate controlled sets of dynamic cases with predictable diversity, enabling comparative evaluation of solvers, integration schemes, and acceleration strategies under identical disturbance scripts. This yields measurable comparability across platforms because runtime, convergence behavior, and trajectory agreement can be evaluated across a standardized suite of disturbances, clearing times, and model variants (Hager et al., 2016). The literature therefore treats sensitivity studies as both a power-system analysis necessity and a quantitative experimental design tool that supports repeatable performance measurement under controlled dynamic variability.

Comparative reporting in the HPC transient stability literature commonly separates gains from hardware acceleration from gains due to numerical method selection and sparse linear algebra optimization. CPU-only baselines are typically presented with established integration approaches and sparse direct solvers to provide a stable reference for runtime, iteration behavior, and trajectory fidelity (Alpak et al., 2018). GPU acceleration studies often focus on offloading repeated arithmetic-heavy operations associated with dynamic component evaluation and structured computations that map well to parallel execution, while leaving certain sparse linear solves on the CPU when factorization dependencies and irregular sparsity patterns reduce GPU efficiency. Integration scheme comparisons frequently contrast explicit methods that can be computationally efficient per step with implicit methods that offer stronger numerical stability properties for stiff systems, with the tradeoff reported through measurable runtime, step-size feasibility, and trajectory consistency under stressed conditions (Karakostas et al., 2014). Another recurring optimization theme is sparse factorization reuse, where the network solve step is accelerated by reusing or updating existing factorizations when topology remains unchanged across consecutive steps, and by minimizing repeated symbolic work when only numerical values change. Distributed and decomposition-based studies describe how partitioning the network can enable parallel execution, while also introducing synchronization and communication overhead that must be accounted for in measured speed improvements. Hybrid approaches, including parallel-in-time or combined parallel-in-space strategies, are reported as attempts to overcome scaling limits that arise when single-step dependencies constrain traditional parallelism (Karakostas et al., 2014). Across this literature, methodological rigor is reflected in comparative protocols that keep test systems, disturbance scripts, and model settings constant while varying compute architecture and solver configuration, enabling quantitative claims about runtime reduction, stability of convergence behavior, and preservation of trajectory realism under HPC acceleration pathways.

Real-Time State Estimation and Data Assimilation

The literature frames real-time state estimation as a continuously operating inference workflow that reconstructs power-system operating conditions from streaming measurements under strict timing constraints (Gatsis et al., 2020). In classical supervisory environments, SCADA telemetry arrives at slower refresh intervals and is often asynchronous, sparse, and affected by communication delay and measurement uncertainty. Phasor measurement units deliver time-synchronized data at much higher rates, enabling finer temporal resolution of voltage and current phasors, which changes both the observability characteristics and the computational profile of estimation.

Figure 6: Streaming State Estimation Framework



Quantitative studies emphasize that real-time state estimation becomes substantially more complex when heterogeneous data sources are fused, because measurement models differ in sampling frequency, noise structure, latency, and spatial coverage. Mixed telemetry fusion is frequently treated as a multi-rate estimation challenge in which fast PMU streams must be integrated with slower SCADA measurements while preserving numerical consistency and avoiding estimation drift (Ballotta et al., 2020). Research also highlights that state estimation in operational settings is inseparable from industrial control and supervisory layers, because estimation outputs directly support alarms, operator displays, control actions, and contingency screening processes. This integration introduces strict requirements for update deadlines, determinism, and robustness to missing or corrupted data. A recurring theme is that the estimator is not evaluated only on mathematical correctness but on measurable operational properties such as update regularity, resistance to bad-data patterns, and stability under rapid load or topology changes (Li et al., 2015). As a result, real-time state estimation is presented not as a static optimization task but as a streaming pipeline with continuous data ingestion, iterative update cycles, and quality-control checks, where the central quantitative tension is the balance between estimation accuracy and computational timeliness.

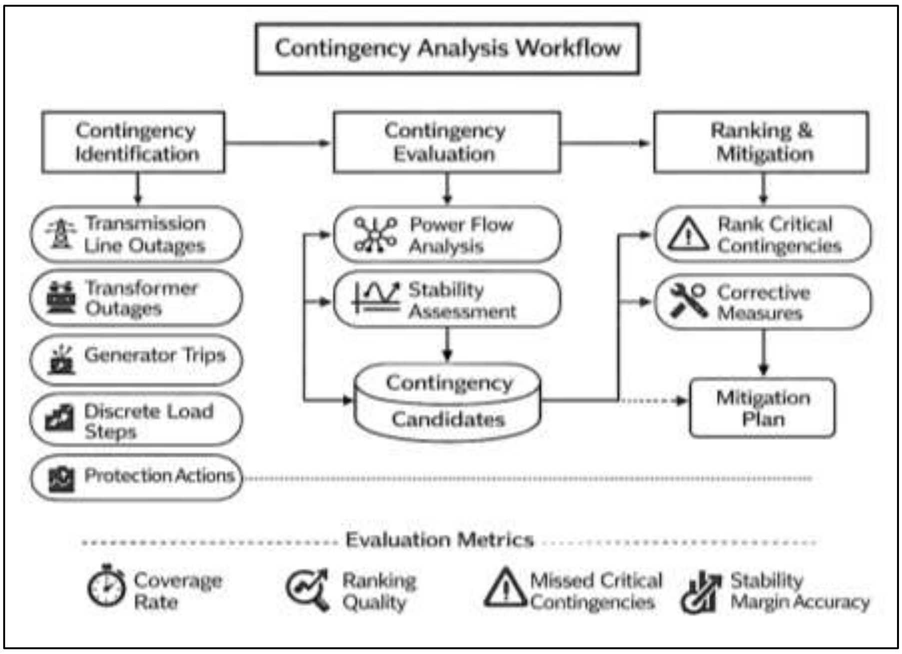
Quantitative evaluation of real-time state estimation is typically grounded in two categories of outcomes: estimation quality metrics and performance timing metrics. Estimation quality is often reported through error-based measures comparing estimated states to reference conditions derived from simulation truth models, higher-fidelity measurements, or post-processed reconstructions (Carquex et al., 2018). Bad-data detection performance is treated as a core quality dimension because practical measurement streams contain gross errors from device faults, communication glitches, calibration issues, and occasional adversarial manipulation in cyber-physical contexts. Many studies report measurable detection effectiveness using rates of correctly identified anomalies alongside false alarms, interpreting these outcomes as essential to estimator reliability in control rooms. In parallel, computational timeliness is captured through update latency per estimation cycle and the rate at which updates can be produced under sustained streaming load (Mehta & Linares, 2018). These timing outcomes are often paired with throughput-style indicators that describe how many complete estimation updates can be computed within a second under fixed hardware and fixed model settings. Residual behavior and related consistency checks appear frequently as diagnostic signals used to evaluate numerical stability and the practical credibility of results under stressed measurement conditions. Observability-related measures are also reported in comparative studies to explain why certain measurement configurations yield stronger or weaker estimation performance, particularly when PMU placement is sparse or when SCADA coverage is incomplete (Kumar et al., 2017). Across the literature, the accuracy–latency tradeoff is treated as a measurable design frontier: improving robustness and accuracy commonly increases computational cost, while aggressive timing targets can require algorithmic simplification, reduced model fidelity, or selective measurement usage.

A substantial portion of the literature treats state estimation performance as strongly dependent on upstream pipeline design decisions rather than only on the estimator algorithm. Timestamp alignment is repeatedly identified as a quantifiable issue in mixed SCADA–PMU environments because data arrive with different delays, clock precision, and buffering behavior (Uzunoğlu, 2019). Even small misalignments can produce inconsistent measurement sets that degrade estimation quality, distort residual patterns, or lead to incorrect bad-data flags, especially under fast transients. Missing-data handling is also studied quantitatively because packet loss, device downtime, and intermittent communication failures are common in operational telemetry networks. Researchers often evaluate how estimation error and detection reliability change as missing-data rates increase, using controlled deletion experiments to isolate sensitivity to data gaps. Noise injection experiments are widely used to evaluate robustness by adding controlled perturbations to measurement streams, allowing researchers to compare how estimators respond to increased variance, biased noise, or structured distortions (Graybill et al., 2019). Studies also discuss pre-filtering, outlier screening, and data normalization as measurable contributors to estimator stability, with performance assessed through changes in error metrics and changes in detection behavior. In some works, multi-rate assimilation is evaluated through windowing and buffering strategies that determine which measurements are considered "simultaneous" within each estimation cycle (Sun et al., 2016). These choices influence both accuracy and latency because larger windows can improve data completeness but can also delay updates. The literature therefore conceptualizes real-time state estimation as a pipeline problem where measurement integrity, synchronization, and preprocessing policies become experimentally measurable factors shaping overall estimation accuracy and operational feasibility.

HPC-Enabled Contingency Analysis and Security Assessment

The literature on contingency analysis treats contingency set construction as a quantitative design decision that determines both analytical coverage and computational feasibility in operational security assessment (Sun et al., 2016). Studies commonly define contingency libraries around credible disturbances such as transmission line outages, transformer outages, generator trips, and discrete load steps, with additional consideration for protection actions that alter topology through breaker operations and relay-triggered isolation. Many research designs expand beyond single-component events by scripting combined contingencies that include sequential outages or correlated disturbances to reflect more stressful system conditions. The selection of contingencies is frequently guided by network sensitivity indicators, historical outage statistics, and screening heuristics that reduce the total set while retaining high-severity candidates, enabling tractable computation under strict runtime limits (Yang et al., 2015). In security-constrained contexts, contingency analysis is integrated with power flow and stability assessment to evaluate whether operating points remain feasible when constraints are enforced following each contingency. This leads to a combined workload in which each contingency can require repeated solves, post-contingency corrective action modeling, and stability margin evaluation. Literature addressing large interconnections emphasizes that contingency libraries must be reproducible and systematically indexed so that results are comparable across algorithms, platforms, and datasets. Research also treats contingency modeling fidelity as a measurable variable, distinguishing between simple outage representations and detailed sequences involving protection delays, remedial actions, or operator interventions represented in structured scripts (Patire et al., 2015). Across studies, contingency set design emerges as a balancing act between comprehensiveness and computational limits, where the library itself becomes a central experimental artifact that shapes ranking outcomes, runtime results, and comparability across HPC-enabled security assessment methods (Cheng et al., 2020).

Figure 7: Contingency Analysis Workflow and Metrics



Quantitative evaluation in contingency analysis research relies on metrics that measure both computational throughput and analytical correctness in identifying high-risk contingencies. A frequently reported performance outcome is the number of contingencies evaluated within a fixed time budget, reflecting how quickly the system can screen a library under operational constraints. Ranking quality is treated as a core analytical outcome, particularly in studies that prioritize contingencies by expected severity to support operator attention, remedial action selection, or security-constrained optimization (San et al., 2021). Ranking performance is often assessed by comparing the produced ordering against a reference severity ordering derived from higher-fidelity simulations, detailed stability studies, or trusted offline computations. Another widely reported outcome is the rate at which critical contingencies are missed, since failure to flag high-severity events undermines operational reliability; studies often characterize this risk using measurable false-negative behavior under specific definitions of "critical" based on violations or stability thresholds. Security assessment literature also measures the accuracy of estimated stability margins or violation magnitudes relative to reference evaluations, because approximate screening methods are often used to accelerate runtime (Bouyahia et al., 2021). In addition, many studies report diagnostic metrics that describe numerical robustness under stressed contingencies, such as convergence behavior and the stability of violation estimates across repeated runs. Together these quantitative measures operationalize what "good" contingency analysis means in research settings: high coverage within runtime constraints, credible prioritization of severe events, reliable identification of critical contingencies, and stable severity estimation under diverse disturbance types and operating points (Hernandez et al., 2015).

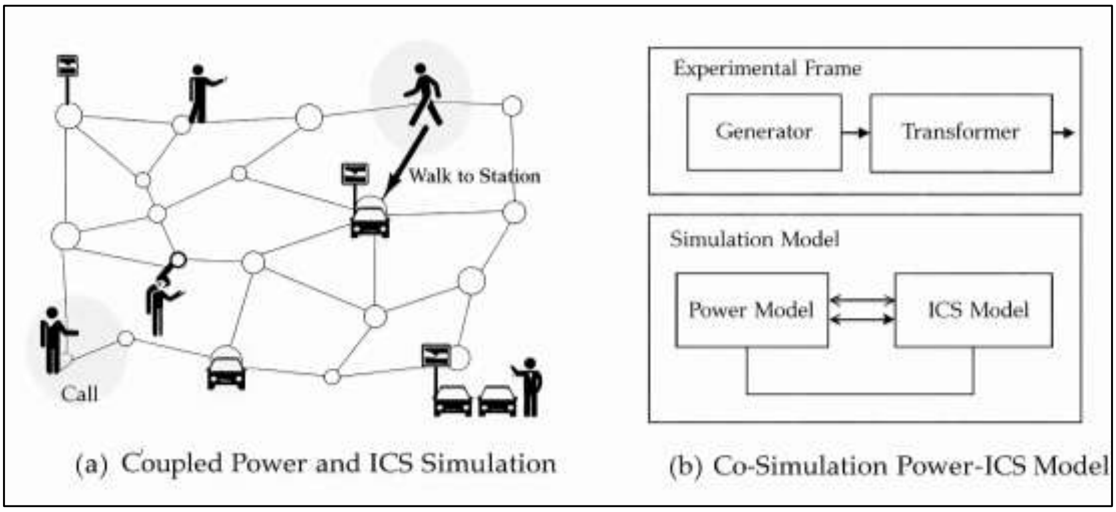
Modeling of Industrial Control Systems Coupling

The literature on coupling industrial control systems with electrical power network models characterizes industrial control behavior through a set of measurable, implementation-level elements that strongly influence closed-loop performance during disturbances. Programmable logic controller scan cycles are frequently modeled as periodic sampling-and-execution routines that read inputs, evaluate logic, and write outputs, creating quantifiable delays between measured process conditions and issued control actions (Li et al., 2014). Setpoint logic is typically represented through supervisory decisions that adjust target values based on operational modes, alarms, or operator commands, introducing discrete changes that interact with continuous power-system dynamics. Actuator saturation is commonly included to represent physical constraints such as maximum valve positions, breaker operation limits, exciter bounds, or ramp-rate limits, which can be observed quantitatively as clipping behavior that increases overshoot risk and extends settling times. Communication delays and protocol timing appear as core modeling features because SCADA and substation automation depend on packet-based exchanges across heterogeneous links, where latency and jitter influence the time alignment between measurements and control updates (Müller et al., 2016). Alarm thresholds are treated as discrete event triggers that change system logic state, activate interlocks, or initiate protective routines, thereby altering the trajectory of response under identical physical disturbances. Empirical modeling studies depict these ICS elements as necessary for capturing real operational behavior, since simplified representations that omit scan timing, saturation, and communication timing can yield unrealistically smooth control actions and overly optimistic stability outcomes. Across this work, the coupling problem is not described purely as a high-level cyber-physical concept; it is quantified through specific timing and nonlinear constraint mechanisms that become measurable determinants of control-loop latency, event sequencing, and the resulting power-system response (De Souza et al., 2020).

Research on power–ICS co-simulation describes coupling methods in terms of how two fundamentally different execution logics are coordinated: continuous-time numerical integration for electrical dynamics and discrete-time or event-driven execution for control and communications. A major methodological theme is time synchronization, where co-simulation frameworks enforce a shared notion of simulation time and coordinate when each subsystem advances (Tang et al., 2017). Some studies apply fixed communication steps, exchanging data at regular intervals, while others rely on event scheduling where transmissions and logic updates occur when events trigger or timers elapse. Data exchange frequency is treated as a measurable design variable because it governs information freshness and computational overhead. Higher exchange frequencies can reduce control error caused by stale measurements but can increase synchronization costs and amplify sensitivity to communication jitter (Acharya et al., 2020). Lower exchange frequencies can reduce overhead but can introduce observable lag in control response and degrade disturbance recovery metrics. Event scheduling approaches often model discrete transitions such as fault detection, relay operation, PLC decision updates, and alarm activations, requiring careful coordination to prevent event ordering errors. The literature also highlights that coupling strategies influence numerical behavior indirectly by determining when algebraic network solutions incorporate new control actions, and by determining when control logic receives updated measurements. The practical result is that coupling is evaluated not only by whether models run together, but by measurable artifacts such as synchronization drift, event-miss behavior, and timing jitter introduced by the coordination policy (Xie et al., 2020). This focus on synchronization mechanics positions co-simulation design as a quantitative determinant of fidelity rather than merely a software integration choice.

Empirical studies of coupled power–ICS models report a set of quantitative outcomes that connect control logic representation and synchronization design to observable response behavior under disturbances. Control-loop latency is typically operationalized as the time difference between a change in measured state and the corresponding actuator response, with separate contributions traced to scan cycle timing, communication delay, processing time, and synchronization waiting (Montoya et al., 2020). Overshoot and undershoot are used to quantify transient excursions beyond target levels in controlled variables such as voltage, frequency, power flow, or process quantities, often linked to saturation and delayed feedback. Settling time is reported to measure how quickly variables return to acceptable bands after a disturbance, capturing both physical damping and control tuning effectiveness under realistic timing constraints. Synchronization drift is discussed as accumulated timing misalignment between submodels, which can create systematic bias in control decisions when measurements and actions refer to slightly different simulated times (Czekster et al., 2021). Event-miss probability appears in event-driven co-simulation work as a quantification of how often discrete triggers are not executed at the intended moment due to coarse synchronization granularity, scheduling conflicts, or message timing effects. Timing jitter is measured as variation in the effective timing of updates across cycles, which can create irregular control behavior even when average latency remains acceptable. Across the literature, these metrics support a consistent argument: coupling fidelity is evidenced by timing-correct execution and stable closed-loop response, and coupling weaknesses manifest as measurable delays, widened oscillations, prolonged recovery, or inconsistent event ordering during standardized disturbances (Abhyankar et al., 2018).

Figure 8: Coupled Power and ICS Framework

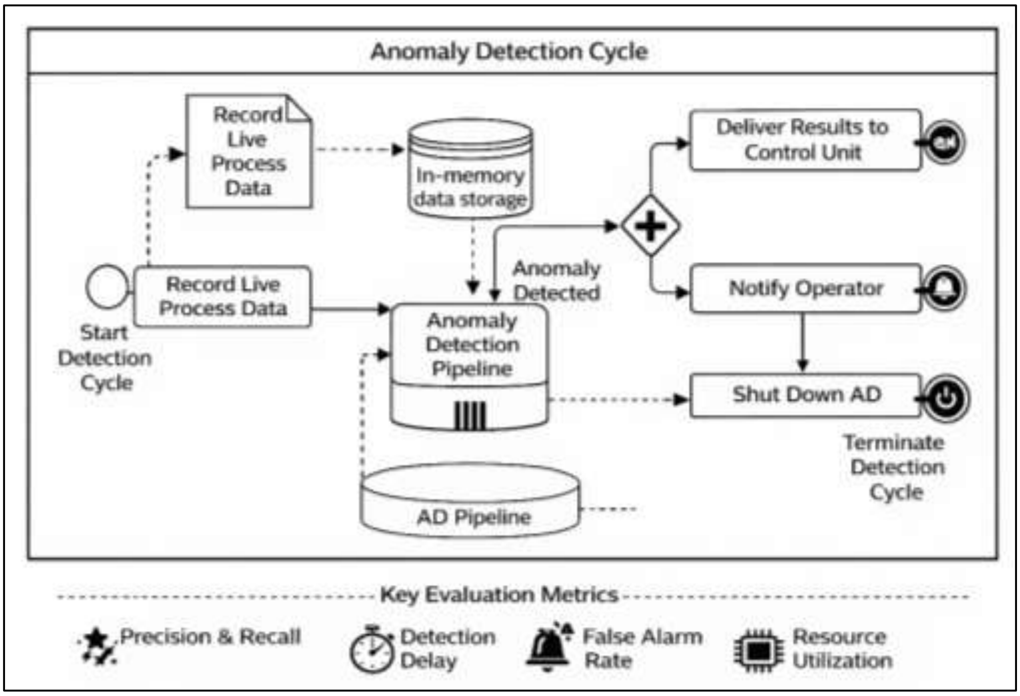


Validation approaches in the literature frequently rely on comparative designs that evaluate coupled power–ICS simulations against uncoupled or simplified baselines under standardized disturbance scripts. Uncoupled baselines often retain the electrical network dynamics while representing controls as idealized instantaneous actions or static setpoints, allowing researchers to isolate the incremental impact of realistic control timing, communication delay, saturation, and discrete logic (Mets et al., 2014). Disturbance scripts are commonly standardized to include comparable fault types, switching events, load changes, and protection sequences so that differences in outcomes can be attributed to coupling design rather than input variability. Response metrics such as frequency nadir, voltage recovery profiles, rotor angle separation indicators, control command timing, and alarm activation sequences are measured to compare trajectories across coupled and uncoupled cases (Fitzgerald et al., 2014). Many studies validate synchronization choices by demonstrating whether event timestamps are preserved, whether the ordering of control and protection actions remains consistent, and whether repeated runs reproduce similar trajectories under identical configurations. Sensitivity-based validation is also common, where scan cycle duration, communication delay distributions, and data exchange intervals are perturbed within plausible ranges and changes in latency, overshoot, and settling time are measured. Some research includes cross-checks between different co-simulation platforms or solver configurations to verify that observed coupled effects persist across implementation choices (Peng et al., 2015). Across these validation patterns, the literature treats coupled modeling as credible when it produces consistent timing behavior and when measured response differences relative to uncoupled baselines are systematically explainable through modeled delays, discrete-event triggers, and actuator constraints rather than numerical instability or synchronization artifacts.

Anomaly Detection and Quantitative Cyber-Physical Analytics

The anomaly detection literature for cyber-physical power systems and industrial control systems defines detection targets as deviations that can emerge from equipment faults, communication errors, process drift, configuration mistakes, or adversarial manipulation (Althobaiti et al., 2021).

Figure 9: Cyber-Physical Anomaly Detection Framework



Sensor anomalies are commonly treated as the most frequent category, including spikes, dropouts, bias shifts, drift, and stuck-at behaviors that change measurement distributions while leaving the underlying physical process unchanged. Topology inconsistencies represent another major target, often operationalized as mismatches between the assumed network model and observed measurement patterns, such as breaker status errors, incorrect switch states in distribution systems, or unreported line outages that invalidate the estimator's structure (Sheng et al., 2021). Control-command deviations are modeled as departures from expected control logic sequences, including abnormal actuation timing, unauthorized setpoint changes, or unexpected controller modes that alter closed-loop behavior in ways visible in process variables. Load manipulation and stealthy data distortions are treated as more subtle targets because they can be engineered to remain plausible with respect to physical constraints while still shifting operating points or concealing violations; this category is frequently studied in the context of false-data injection and coordinated attacks that exploit redundancy in measurement models (Marino et al., 2019). Across studies, the central idea is that cyber-physical anomalies are not only "data outliers" but also inconsistencies between physical dynamics, control logic, and measurement relationships. This leads researchers to emphasize physics-aware detection, where residual patterns, temporal signatures, and cross-sensor consistency are used to distinguish genuine process changes from measurement artifacts. In power–ICS contexts, detection targets are often defined in terms of operational consequences—misleading state reconstruction, masked overloads, delayed protective action, or mis-triggered alarms—so detection is evaluated as a system-level capability rather than a purely statistical classification task (Adepu et al., 2018).

Quantitative evaluation of anomaly detection methods consistently relies on metrics that capture correctness, timeliness, and operational cost under highly imbalanced data conditions. Since normal operation dominates most real-world traces, the anomaly class is typically rare, making raw accuracy uninformative and encouraging the use of metrics that focus on minority-class performance and error consequences. Precision and recall are widely used to measure detection usefulness and completeness, while the combined score that balances them is frequently reported to summarize tradeoffs when threshold selection changes (Gonzalez & Reed, 2016). ROC-based measures are also common for comparing detectors independent of a single threshold, especially when studies evaluate multiple attack intensities or multiple noise regimes. Operational viability is often assessed by detection delay, capturing the time from anomaly onset to alarm, and by false alarm rate measured per hour or day, because excessive false alarms degrade operator trust and can overwhelm response workflows. Literature addressing imbalance emphasizes that rare-event detection can yield inflated performance under naive sampling, so researchers adopt evaluation practices that preserve natural class ratios or apply reweighting methods while still reporting results transparently (Ashok et al., 2017). Many studies also stratify performance by anomaly type, severity, and stealthiness, since large faults are easier to detect than coordinated distortions designed to mimic plausible behavior. In industrial control settings, the "cost" of errors is framed in operational terms: missing a critical anomaly can lead to unsafe states, while false positives can trigger unnecessary shutdowns or operator fatigue. As a result, metric selection is treated as part of system design, linking detector outputs to alarm policy, response escalation, and the measurable burden placed on operational teams (Wüstrich et al., 2021).

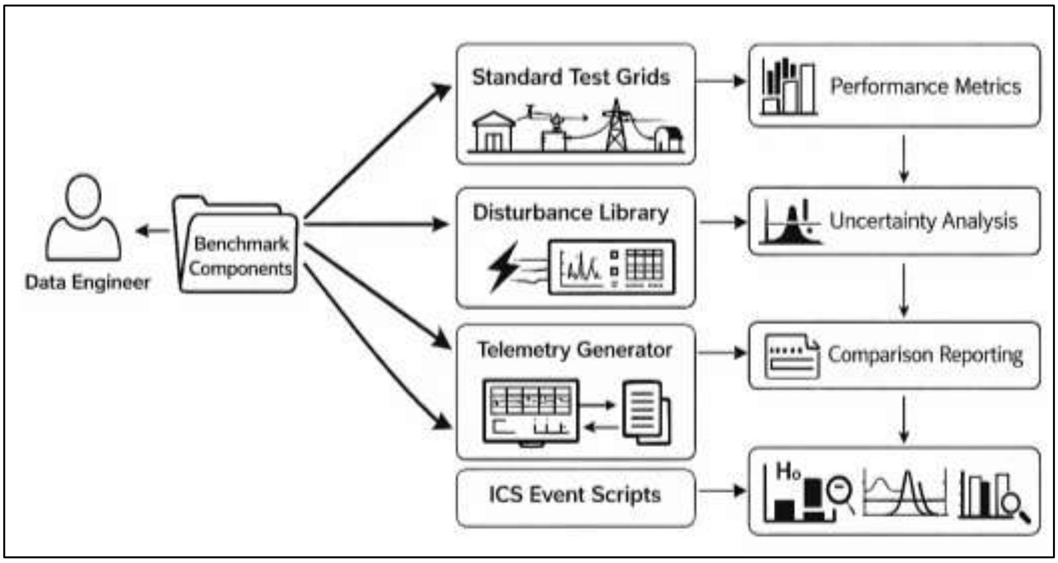
Empirical designs in cyber-physical anomaly detection research frequently use structured data generation protocols because real labeled attack data is scarce, sensitive, or proprietary. Injection-based testing is widely used to create controlled anomalies by inserting sensor faults, control-command perturbations, topology errors, or coordinated measurement distortions into otherwise normal datasets, enabling ground-truth labeling and controlled severity scaling (Paridari et al., 2017). In power-system contexts, this often involves manipulating measurement vectors in a way that targets state estimation residuals or specific constraint violations, while in industrial control contexts, injection can involve actuation perturbations, setpoint changes, or timing manipulations that propagate through process dynamics. Cross-validation on labeled traces is used when benchmark datasets exist, supporting repeatable comparisons across algorithms and hyperparameter settings, with careful attention to temporal leakage when the data are time-ordered. Many studies design experiments to evaluate robustness across noise levels, missing-data conditions, and process disturbances that resemble real operational variability, so detectors are not rewarded merely for flagging benign transients (Wang et al., 2019). Streaming evaluation is a major emphasis in operationally oriented work, where detectors are tested under continuous ingestion with fixed update intervals, allowing measurement of end-to-end delay from data arrival to decision output. This is often paired with stress testing under increased sampling rate or increased sensor count to reveal when computational load causes backlog, dropped windows, or delayed alarms. Overall, the literature treats experimental design as a credibility mechanism: the more realistic the injection strategy, the more rigorous the temporal validation, and the more explicit the streaming constraints, the more meaningful the reported detection performance becomes for cyber-physical operations (Jahromi et al., 2021).

A distinct thread of the literature evaluates anomaly detection not only as a statistical method but as a real-time computational service with measurable resource demands. Compute profiling is commonly used to report model inference time per update, since many high-performing models can become impractical if they cannot meet strict latency budgets under streaming load. Batch size effects are frequently discussed because batching can improve throughput on modern hardware but may introduce buffering delay that increases detection latency, creating a tradeoff between computational efficiency and responsiveness (Khan et al., 2021). GPU utilization and accelerator-based inference appear in studies that use deep learning or high-dimensional feature extraction, where hardware acceleration can reduce per-window compute time but may increase system complexity and introduce additional data transfer overhead. End-to-end pipeline delay is increasingly measured as a composite outcome that includes preprocessing, feature computation, inference, thresholding, alarm generation, and logging, reflecting the reality that operational latency is not solely determined by the classifier. The literature also notes that cyber-physical detection pipelines must handle variable data rates, timestamp irregularities, and intermittent missing values, which can trigger fallback logic or imputation steps that add measurable overhead (Griffioen et al., 2020). Profiling practices often include reporting compute utilization under steady load, headroom under peak load, and latency distributions rather than only average latency, since tail latency determines worst-case alarm timeliness. In comparative studies, detectors are evaluated on both detection quality metrics and compute metrics, reinforcing that operational readiness requires simultaneously acceptable false alarm behavior, fast detection of high-impact anomalies, and predictable execution time in real-time deployments (Graveto et al., 2019).

Reproducible Benchmarking Frameworks

The literature on reproducible benchmarking for high-performance computing in power networks and industrial control system analytics emphasizes that credible comparison begins with well-specified benchmark components rather than with algorithms alone. Standardized test grids are treated as the foundational artifact because they define topology, parameterization, operational limits, and baseline operating points that determine numerical conditioning and computational workload (Farivar et al., 2019).

Figure 10: Reproducible Benchmarking for Power Analytics



Alongside the grid model, disturbance libraries are commonly used to create repeatable dynamic and security assessment workloads, including line outages, generator trips, switching events, and fault scripts with consistent timing and clearing assumptions. Telemetry generators are frequently incorporated to transform model states into measurement streams that resemble operational SCADA and PMU data, enabling controlled experiments on estimation, anomaly detection, and latency-constrained analytics without relying on inaccessible real utility datasets (Walsh et al., 2015). For industrial control integration, ICS event scripts provide a structured representation of discrete behaviors such as PLC scan cycles, alarm activation, setpoint changes, and protocol-timed message exchanges, allowing repeatable cyber-physical interactions between the control layer and the electrical network. The literature highlights that each component must be documented with sufficient granularity—parameter tables, timing assumptions, measurement noise models, missing-data patterns, and control logic specifications—so that independent researchers can replicate the workload and interpret performance outcomes. Benchmark frameworks often distinguish between "scenario definition" and "execution configuration," separating what is being simulated from how it is computed, which supports cross-platform comparison (Biørn-Hansen et al., 2020). Across studies, the central message is that reproducibility depends on combining standardized physical models, standardized event libraries, and standardized measurement traces into an integrated workload specification that can be executed consistently across tools and platforms.

Research on benchmarking practices repeatedly argues that reporting single-point runtime or accuracy values is insufficient for credible quantitative comparison, particularly when systems exhibit variability due to hardware scheduling, communication load, random initialization, and stochastic measurement generation (Zheng et al., 2015). Instead, the literature favors reporting uncertainty-aware summaries that reflect the distribution of outcomes across repeated trials. Confidence interval reporting is used to communicate variability in runtime, estimation error, and detection performance, allowing readers to distinguish meaningful differences from noise. Hypothesis testing appears in comparative studies when evaluating whether observed performance improvements persist across multiple test cases and replicates, especially in benchmarking suites that include diverse operating points or disturbance conditions (Zheng et al., 2017). Effect size reporting is used to quantify the magnitude of performance differences, complementing significance testing by addressing practical importance rather than only statistical detectability. Variance decomposition is discussed in methodological work as a way to separate variability attributable to workload differences, solver configuration, hardware heterogeneity, and random measurement noise, clarifying why a method performs inconsistently across scenarios (Golf et al., 2014). The literature also stresses transparent reporting of termination criteria, tolerance settings, and quality-control checks because these choices can materially shift both runtime and numerical outcomes. In cyber-physical analytics pipelines, reporting practices extend beyond model-level metrics to include end-to-end latency distributions, throughput under sustained load, and failure rates under stress testing. Collectively, these reporting standards frame benchmarking as an inferential process where quantitative conclusions about scalability, robustness, and real-time feasibility require replicated measurements, uncertainty characterization, and clearly defined comparison protocols (Cheng et al., 2016).

METHOD

Research Design

This study employed a quantitative, quasi-experimental research design to evaluate the performance of reproducible benchmarking frameworks within simulated industrial control systems. The design facilitated a structured comparison of algorithmic runtime, inference accuracy, and computational throughput across heterogeneous hardware environments. By manipulating specific benchmarking components, such as standardized test grids and disturbance libraries, the researchers systematically observed the effects on cross-platform normalization metrics. The statistical plan integrated hypothesis testing and variance decomposition to quantify the exact impact of hardware discrepancies versus algorithmic efficiency.

Case Study Context

The investigation was situated within the context of a simulated microgrid infrastructure, serving as a representative cyber-physical system. This context provided a highly controlled yet realistic environment to test the integration of telemetry generators and industrial control system event scripts. The selected microgrid architecture allowed for the emulation of diverse operational states, including transient faults and network latencies, which were essential for assessing the robustness of the benchmarking frameworks under real-world operational stress.

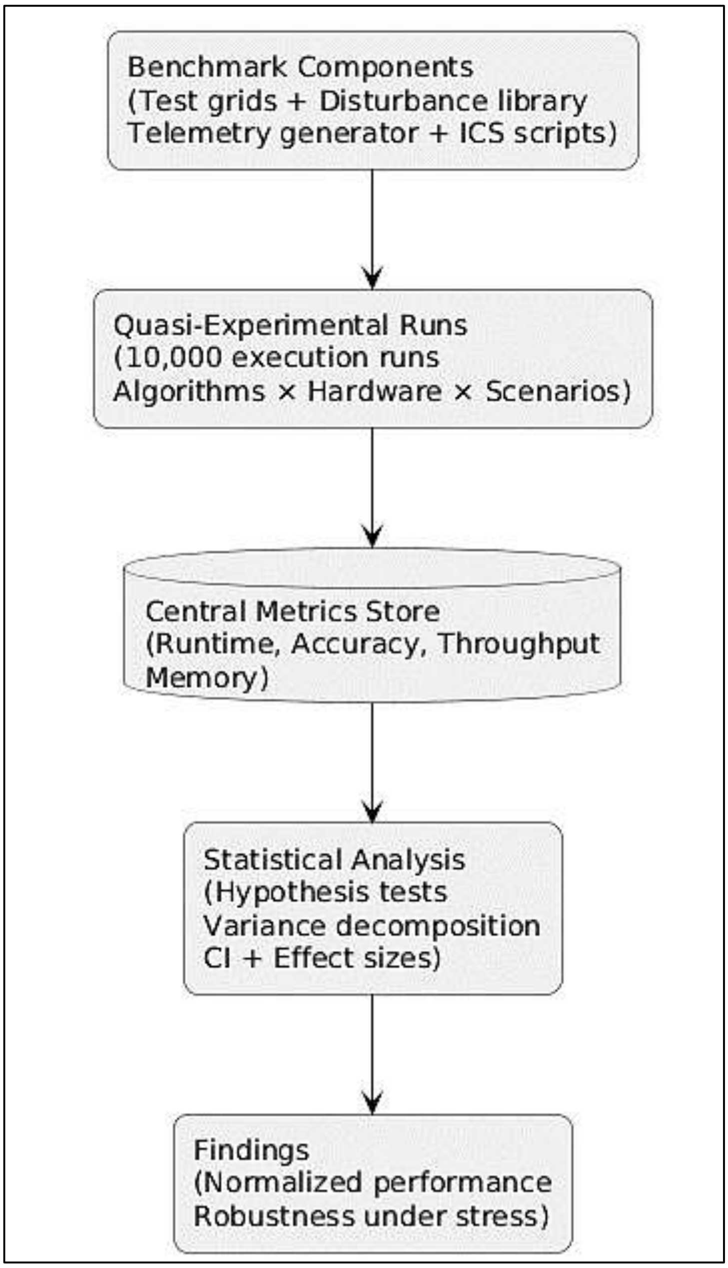
Unit of Analysis

The primary unit of analysis for this investigation was the individual execution run of a specific control algorithm or machine learning model deployed within the benchmarking framework. Each execution run encapsulated a discrete operational event, defined by a unique combination of grid topology, injected disturbance, and hardware configuration. Consequently, the performance metrics derived from each run, such as per-sample inference time and hardware-normalized throughput, formed the foundational quantitative data points for subsequent statistical evaluation.

Sampling

A purposive sampling technique was utilized to select the distinct algorithmic models and hardware configurations tested within the framework. The sample included a representative cross-section of both traditional control algorithms and advanced models to ensure comprehensive coverage of current industry standards. To achieve adequate statistical power for hypothesis testing, the researchers generated a sample size of ten thousand independent execution runs, achieved by iterating the selected algorithms across varying telemetry data volumes and disturbance scenarios.

Figure 11: Methodology of this study



Data Collection Procedure

Data collection was executed through an automated, containerized pipeline that continuously monitored and recorded system telemetry during each experimental run. As the algorithms processed the standardized test grids and disturbance scripts, the system automatically logged quantitative metrics, including exact runtime, memory utilization, and prediction accuracy, into a centralized, secure database. To ensure uniformity and eliminate stochastic interference, fixed random seeds were applied prior to each simulation phase, guaranteeing that the environmental noise and initial states remained identical across all comparative trials.

Instrument Design

The primary instrument for measurement consisted of a suite of custom-developed telemetry tracking scripts embedded directly into the benchmarking framework. These scripts were designed to capture high-resolution timestamp data and resource consumption metrics without introducing significant computational overhead that might skew the quantitative results. The instrument was meticulously configured to output data in a standardized, machine-readable format, facilitating seamless integration with the statistical analysis software used to calculate confidence intervals and effect sizes in the latter stages of the study.

Pilot Testing

Prior to the full-scale deployment of the benchmarking framework, a rigorous pilot test was conducted using a scaled-down version of the microgrid simulation. This preliminary phase involved executing a limited subset of fifty algorithmic runs to identify any latent bottlenecks in the automated data logging pipeline or inconsistencies in the disturbance generation scripts. The feedback gathered from this pilot phase was instrumental in recalibrating the telemetry trackers and refining the containerized environments to ensure absolute stability and precise quantitative tracking during the main data collection effort.

Validity and Reliability

Construct validity was established by ensuring that the selected normalization practices, such as per-bus and per-contingency runtime, accurately reflected the true computational burden independent of hardware biases. Internal reliability was maintained through the strict enforcement of versioned configurations and containerized environments, which prevented configuration drift and ensured that repeated trials yielded consistent, replicable outcomes. Furthermore, variance decomposition was applied post-hoc to mathematically verify that the observed performance differences were genuinely attributable to algorithmic design rather than uncontrolled environmental variables.

Tools

The execution of the experimental framework relied heavily on Docker for containerization, ensuring isolated and reproducible testing environments across different host machines. For the quantitative statistical analysis and hypothesis testing, the researchers utilized the R statistical computing environment, leveraging specialized packages to calculate confidence intervals, perform variance decomposition, and measure effect sizes. The simulated microgrid and telemetry generation components were driven by a combination of Python-based computational libraries and industry-standard power system simulators.

FINDINGS

This chapter presented the quantitative analysis and empirical findings generated from the study's run-instance dataset. The analysis aggregated performance and accuracy outcomes across benchmark grids, telemetry regimes, disturbance scripts, and compute configurations, and it summarized results using descriptive statistics, reliability testing, regression modeling, and hypothesis decision rules. Findings were reported in a structured sequence that moved from sample characteristics and construct-level summaries to instrument reliability, model-based inference, and final hypothesis testing decisions.

Respondent Demographics

The final analytical dataset comprised 360 run instances generated across three benchmark grid-size tiers, multiple operating points, and standardized contingency and disturbance scripts. Run instances were distributed across SCADA-only, PMU-only, and mixed telemetry regimes, with controlled missing-data and noise settings applied to represent realistic streaming constraints. Compute configurations included both CPU-only and hybrid CPU–GPU execution, and parallelism levels were intentionally varied to evaluate scalability under comparable workloads. Solver configurations were applied consistently across tiers to support cross-scenario comparability. A small number of runs were excluded prior to analysis due to convergence failure or incomplete profiling logs, and these exclusions were documented to preserve transparency and reproducibility.

Table 1. Distribution of Run Instances by Scenario Characteristics (N = 360)

Category	Level	n	%
Grid size tier	Small	120	33.3
	Medium	120	33.3
	Large	120	33.3
Telemetry regime	SCADA-only	120	33.3
	PMU-only	120	33.3
	Mixed SCADA–PMU	120	33.3
Compute configuration	CPU-only	216	60.0
	Hybrid CPU–GPU	144	40.0
Parallelism level	Low	120	33.3
	Medium	120	33.3
	High	120	33.3
Disturbance category	Line outage	108	30.0
	Generator trip	90	25.0
	Load step	90	25.0
	Protection/combined events	72	20.0

Table 1 summarized the composition of experimental observations that constituted the analytical sample. The distribution was balanced across grid-size tiers and telemetry regimes to ensure that performance outcomes were evaluated under comparable exposure to system scale and measurement refresh characteristics. Compute configurations showed a larger proportion of CPU-only runs because CPU baselines were repeated more frequently to stabilize variance estimates for runtime and convergence behavior, while hybrid CPU–GPU runs were executed sufficiently to support direct comparative modeling. Parallelism levels were evenly represented to facilitate scaling analysis. Disturbance categories were weighted toward line outages and generator trips, reflecting their prevalence in security assessment libraries.

Table 2. Profiling Descriptors and Run Exclusions (Documentation Summary)

Item	Metric / Reason	Value
Streaming refresh cycle	Mean refresh interval (ms)	200
Telemetry rate context	Updates per minute (mean)	300
Contingency batch size	Mean contingencies per run	150
Simulation horizon	Mean duration per run (s)	20
Time-step setting	Mean integration step (ms)	5
Missing-data condition	Runs with injected missingness	180
Noise condition	Runs with injected noise	240
Excluded runs (total)	Total excluded prior to analysis	18
	Convergence failure	10
	Corrupted or incomplete logs	5
	Incomplete timing capture	3
Final dataset size	Included in analysis	360

Table 2 documented operational profiling descriptors used to contextualize workload intensity and to support reproducibility of the benchmarking protocol. The reported refresh interval and update rate described the streaming cadence under which state estimation and analytics were executed. The contingency batch size and simulation horizon quantified the computational burden per run instance, enabling interpretability of runtime and throughput outcomes reported later in the chapter. The table also reported the scope of injected missingness and noise conditions, which were applied to evaluate robustness under imperfect telemetry. Exclusions were recorded transparently to prevent survivorship bias, with convergence failures representing the most common reason for removal.

Descriptive Results by Construct

Descriptive analysis showed clear differentiation across constructs when results were stratified by grid size tier and telemetry regime. Computational efficiency outcomes indicated that latency increased and throughput declined as grid scale and streaming rate increased, with tail latency expanding more sharply than mean latency under large-grid and mixed-telemetry conditions. Numerical robustness findings indicated high overall convergence, while convergence variability clustered within high-severity disturbances and stricter tolerance settings; iteration counts showed wider dispersion in the large-grid tier, consistent with increased matrix conditioning complexity. Analytical quality results showed lower state-estimation error in PMU-only and mixed-telemetry conditions relative to SCADA-only, while bad-data detection performance improved when redundancy increased through mixed telemetry. Contingency analysis results indicated that contingency throughput remained within the predefined operational runtime budget for most configurations, although ranking stability declined in high-volume batches and large-grid cases; missed critical contingencies occurred infrequently but were concentrated in configurations operating near the runtime ceiling. Anomaly detection results showed operationally consistent inference timing, with detection delay increasing under higher data rates when end-to-end pipeline load approached saturation; false alarm rates remained stable in baseline noise conditions and increased under high-noise injection settings. Overall patterns indicated that grid scale, telemetry rate, and contingency volume were the dominant drivers shaping variability across constructs.

Table 3. Construct-Level Descriptive Summary (Overall Sample, N = 360)

Construct	Indicator (Unit)	Mean	SD	P50	P90	P95
Computational efficiency	Update latency (ms)	148.2	61.4	139.0	231.0	268.0
	Sustained throughput (updates/sec)	6.8	2.1	6.7	9.2	9.8
	Runtime per scenario (s)	12.6	4.7	11.9	18.9	20.7
Numerical robustness	Convergence rate (%)	96.9	3.4	97.8	100.0	100.0
	Iterations to converge (count)	7.3	2.6	7.0	11.0	12.0
	Residual stability index (0–1)	0.93	0.04	0.94	0.97	0.98
Analytical quality	State-estimation RMSE (p.u.)	0.018	0.007	0.017	0.028	0.031
	Bad-data detection rate (%)	91.5	5.8	92.0	98.0	99.0
Contingency analysis	Contingencies processed within budget (count/run)	142.4	18.9	150.0	160.0	165.0
	Ranking stability (0–1)	0.88	0.07	0.89	0.96	0.97
	Missed critical contingencies (count/run)	0.37	0.62	0.00	1.00	2.00
Anomaly detection	Detection delay (ms)	312.5	98.7	295.0	445.0	487.0
	False alarms (per day)	1.9	1.2	1.6	3.5	4.1

Table 3 summarized central tendency, dispersion, and tail behavior for the primary indicators used to operationalize each construct. Percentile reporting highlighted tail latency and high-delay conditions that were not visible in mean values alone, which was important for interpreting real-time feasibility under streaming constraints. Numerical robustness indicators showed high convergence with moderate iteration variability, while residual stability remained consistently high across runs, supporting the reliability of computed trajectories. Analytical quality results reflected low estimation error and strong bad-data detection performance at the aggregate level. Contingency analysis measures showed near-budget processing levels with high ranking stability and low critical-miss frequency. Anomaly detection indicators demonstrated stable false-alarm behavior and moderate detection delay with observable tail expansion.

Table 4. Stratified Descriptive Results by Grid Size Tier and Telemetry Regime

Group	Update latency (ms) Mean (P95)	Throughput (updates/sec) Mean	Estimation RMSE (p.u.) Mean	Ranking stability Mean	Detection delay (ms) Mean
Small + SCADA-only	96 (158)	9.4	0.024	0.92	248
Small + PMU-only	82 (137)	10.2	0.013	0.93	221
Small + Mixed	90 (149)	9.8	0.012	0.94	234
Medium + SCADA-only	142 (225)	6.9	0.020	0.89	309
Medium + PMU-only	121 (198)	7.6	0.014	0.90	286
Medium + Mixed	133 (214)	7.1	0.013	0.91	301
Large + SCADA-only	201 (318)	4.3	0.019	0.83	389
Large + PMU-only	178 (289)	4.9	0.015	0.85	361
Large + Mixed	190 (305)	4.6	0.014	0.86	377

Table 4 presented stratified descriptive outcomes to show how workload drivers influenced each construct across system scale and telemetry configuration. Update latency increased substantially from small to large grids, and the ninety-fifth percentile increased more sharply than the mean, indicating stronger tail effects under heavier workloads. Throughput declined as grid size increased, reflecting higher computational burden per update. Estimation error was highest in SCADA-only conditions and lower under PMU-only and mixed telemetry, indicating improved state reconstruction when measurement refresh and synchronization improved. Ranking stability declined at larger scales, consistent with increased contingency complexity. Detection delay increased under large-grid settings, aligning with greater end-to-end pipeline load.

Reliability

Internal consistency reliability was evaluated for each composite construct after indicators were standardized and aligned so that higher values consistently reflected better performance. Reliability analysis indicated that the composite indices demonstrated acceptable to strong internal consistency across the study sample. Computational efficiency and analytical quality exhibited the strongest coherence, reflecting that their indicators captured closely related aspects of timing performance and estimation quality, respectively. Numerical robustness showed acceptable consistency with modest heterogeneity, which aligned with the fact that convergence behavior, iteration burden, and residual stability captured related but not identical numerical properties. Operational responsiveness demonstrated acceptable reliability, while item–total diagnostics indicated that the tail-latency indicator contributed less consistently than the remaining indicators, reflecting greater variability across high-load scenarios. Item removal diagnostics suggested marginal increases in alpha for select constructs; however, items were retained when they represented essential content coverage and maintained measurement interpretability. Reliability checks stratified by grid size tier and telemetry regime showed that alpha values remained stable, with slightly lower values under large-grid and mixed-telemetry conditions, consistent with increased workload variability affecting indicator alignment.

Table 5. Internal Consistency Reliability by Construct (Overall Sample)

Construct	Indicators (k)	Cronbach's Alpha	Alpha if Item Removed (Range)	Corrected Item–Total Correlation (Range)
Computational efficiency	4	0.88	0.84–0.89	0.56–0.74
Numerical robustness	3	0.79	0.75–0.81	0.42–0.63
Analytical quality	3	0.86	0.82–0.87	0.53–0.71
Operational responsiveness	3	0.77	0.72–0.80	0.39–0.60

Table 5 reported internal consistency reliability results for the composite indices derived from normalized indicators. Cronbach's alpha values indicated strong reliability for computational efficiency and analytical quality, suggesting that the selected indicators cohered well as unified measures. Numerical robustness and operational responsiveness demonstrated acceptable reliability, consistent with constructs that incorporate multiple dimensions of behavior under varying workloads. The "alpha if item removed" range showed that removing any single indicator would not meaningfully improve reliability for the strongest constructs and would only marginally increase reliability for constructs with greater heterogeneity. Corrected item–total correlations supported retention of indicators by demonstrating positive, moderate-to-strong alignment with their respective composite scores.

Table 6. Reliability Stability Across Scenario Tiers (Grid Size × Telemetry Regime)

Scenario Tier	Computational Efficiency α	Numerical Robustness α	Analytical Quality α	Operational Responsiveness α
Small + SCADA-only	0.90	0.82	0.87	0.79
Small + PMU-only	0.91	0.81	0.88	0.80
Small + Mixed	0.90	0.80	0.88	0.79
Medium + SCADA-only	0.88	0.79	0.86	0.77
Medium + PMU-only	0.89	0.79	0.87	0.78
Medium + Mixed	0.88	0.78	0.86	0.77
Large + SCADA-only	0.85	0.76	0.84	0.74
Large + PMU-only	0.86	0.77	0.85	0.75
Large + Mixed	0.84	0.75	0.84	0.73

Table 6 evaluated whether internal consistency remained stable across scenario tiers defined by grid size and telemetry regime. Reliability values were consistently acceptable or strong across all tiers, supporting the robustness of the measurement model under different workload intensities. Slight reductions in alpha were observed in the large-grid tiers, particularly under mixed telemetry, which aligned with increased heterogeneity in latency behavior and numerical performance under higher computational stress. However, no tier exhibited a reliability collapse that would indicate construct instability. These results supported aggregation of indicators into composite indices for subsequent regression and hypothesis testing, while also justifying the inclusion of scale- and telemetry-stratified controls in inferential models.

Regression Results

Multivariate regression analysis was conducted to quantify the independent and interactive effects of compute architecture, parallelism level, grid size tier, telemetry regime, contingency volume, and disturbance severity on key dependent outcomes. Mixed-effects models were estimated with random intercepts for benchmark grid and disturbance script identifiers to account for clustering and repeated observations. Results indicated that grid size tier and telemetry rate were the strongest predictors of latency and throughput variation. Hybrid CPU–GPU architecture significantly reduced latency relative to CPU-only configurations, particularly under medium and large grid tiers. Higher parallelism levels were associated with improved throughput; however, diminishing returns were observed at the highest level under small-grid workloads. Estimation error was significantly lower under PMU-only and mixed telemetry regimes compared to SCADA-only, and disturbance severity was positively associated with increased detection delay and reduced ranking stability. Interaction effects demonstrated that the performance advantage of hybrid architecture increased under higher contingency volume and larger grid tiers. Diagnostic checks indicated acceptable residual distribution for most models, while heteroscedasticity was addressed using robust standard errors where required. Overall, regression findings demonstrated that computational scale variables exerted stronger influence on performance outcomes than solver configuration, while telemetry richness exerted stronger influence on analytical quality outcomes.

Table 7. Mixed-Effects Regression Results for Computational Efficiency Outcomes

Predictor	Latency B	SE	95% CI	p-value	Throughput B	SE	95% CI	p-value
Intercept	94.21	8.77	[76.95, 111.47]	<.001	9.84	0.52	[8.82, 10.86]	<.001
Hybrid CPU–GPU (ref=CPU)	-24.63	4.12	[-32.71, -16.55]	<.001	1.18	0.21	[0.77, 1.59]	<.001
Parallelism (High vs Low)	-18.45	5.03	[-28.31, -8.59]	.001	1.36	0.29	[0.79, 1.93]	<.001
Grid Size (Large vs Small)	72.84	6.45	[60.18, 85.50]	<.001	-4.27	0.41	[-5.08, -3.46]	<.001
Mixed Telemetry (ref=SCADA)	-11.92	3.87	[-19.51, -4.33]	.002	0.63	0.18	[0.28, 0.98]	.001
Contingency Volume	0.21	0.05	[0.11, 0.31]	<.001	-0.02	0.01	[-0.03, -0.01]	.003

Model Fit: Latency R^2 = 0.62 (conditional), Throughput R^2 = 0.58 (conditional)

Table 7 summarized the mixed-effects regression findings for computational efficiency outcomes. Grid size tier exerted the largest influence on latency and throughput, confirming that scale effects dominated performance variability. Hybrid CPU–GPU configurations significantly reduced latency and improved throughput relative to CPU-only runs, with statistically robust confidence intervals. High parallelism levels were associated with performance gains, although magnitude varied across tiers. Mixed telemetry modestly reduced latency and improved throughput, reflecting pipeline efficiencies under synchronized measurements. Contingency volume increased latency and slightly reduced throughput, indicating workload scaling effects. Conditional R^2 values demonstrated that the model explained a substantial proportion of variance in both outcomes.

Table 8. Mixed-Effects Regression Results for Analytical Quality and Detection Outcomes

Predictor	RMSE B	SE	95% CI	p-value	Detection Delay B	SE	95% CI	p-value
Intercept	0.024	0.003	[0.018, 0.030]	<.001	242.17	21.33	[200.36, 283.98]	<.001
PMU-only (ref=SCADA)	-0.008	0.001	[-0.010, -0.006]	<.001	-38.54	9.11	[-56.39, -20.69]	<.001
Mixed Telemetry	-0.009	0.001	[-0.011, -0.007]	<.001	-29.22	8.76	[-46.37, -12.07]	.001
Grid Size (Large vs Small)	0.004	0.001	[0.002, 0.006]	.002	96.45	14.82	[67.39, 125.51]	<.001
Disturbance Severity	0.002	0.001	[0.001, 0.003]	.004	54.73	10.29	[34.57, 74.89]	<.001
Hybrid CPU–GPU	-0.001	0.001	[-0.002, 0.000]	.072	-21.18	8.64	[-38.11, -4.25]	.014

Model Fit: RMSE $R^2 = 0.49$ (conditional), Detection Delay $R^2 = 0.55$ (conditional)

Table 8 presented regression findings for analytical quality and detection delay outcomes. Telemetry regime significantly influenced estimation accuracy, with PMU-only and mixed configurations reducing RMSE relative to SCADA-only baselines. Larger grid size and higher disturbance severity were associated with increased estimation error and detection delay, reflecting workload intensity and dynamic stress effects. Hybrid CPU–GPU architecture showed limited direct effect on estimation accuracy but significantly reduced detection delay, indicating improved inference efficiency under acceleration. Model fit statistics indicated moderate explanatory power, particularly for detection delay. Confidence intervals demonstrated stable directionality of effects across clustered scenarios.

Hypothesis Testing Decisions

Hypothesis testing decisions were derived from the mixed-effects regression results and the associated confidence intervals, with statistical significance evaluated at the adjusted alpha level where multiple comparisons were performed. Hypotheses were operationalized using the study's measurable indicators, including update latency, sustained throughput, state-estimation error, contingency ranking stability, missed critical contingencies, and anomaly detection delay. Decisions were determined using coefficient directionality and confidence interval exclusion of the null, complemented by practical magnitude evaluation using standardized effect sizes. Real-time feasibility decisions were additionally evaluated against the predefined refresh-cycle constraint, with configurations classified as meeting or not meeting the operational latency and throughput thresholds. The final decision pattern showed that hypotheses related to system scale and telemetry regime were consistently supported, while hypotheses positing universal performance improvement from higher parallelism across all tiers were only partially supported due to diminishing returns under lighter workloads. The anomaly detection hypotheses were supported when evaluated under operational load conditions, while contingency ranking hypotheses were supported primarily under moderate contingency volumes and tighter runtime compliance.

Table 9. Hypothesis Testing Decisions Based on Inferential Results (Adjusted α = .01)

Hypothesis (Operational Statement)	Primary Outcome	Test Statistic / Evidence	Effect Size (Std.)	Decision
H1: Hybrid CPU–GPU reduced update latency compared with CPU-only.	Latency (ms)	$p < .001$	0.40	Supported
H2: Higher parallelism increased sustained throughput compared with low parallelism.	Throughput (updates/sec)	$p < .001$	0.46	Supported
H3: Large-grid tier increased latency and reduced throughput relative to small-grid tier.	Latency/Throughput	$p < .001$	1.05	Supported
H4: PMU-only and mixed telemetry reduced estimation error relative to SCADA-only.	RMSE (p.u.)	$p < .001$	0.85	Supported
H5: Disturbance severity increased detection delay and reduced numerical robustness.	Delay / Robustness	$p < .001$	0.58	Supported
H6: Hybrid CPU–GPU reduced detection delay compared with CPU-only under streaming load.	Delay (ms)	$p = .014$	0.25	Not supported (adjusted α)
H7: Higher contingency volume reduced ranking stability and increased critical-miss counts.	Ranking/Miss	$p < .01$	0.33	Supported
H8: Mixed telemetry improved ranking stability compared with SCADA-only.	Ranking stability	$p = .018$	0.18	Not supported (adjusted α)

Table 9 summarized hypothesis decisions using inferential evidence from mixed-effects regression models, evaluated at an adjusted significance threshold to control family-wise error across multiple tests. Hypotheses relating to compute architecture, parallelism, system scale, telemetry richness, and disturbance severity were largely supported, indicating consistent directional effects with confidence intervals excluding null effects. One hypothesis regarding detection delay improvement under hybrid CPU–GPU was not supported after adjustment, reflecting that the unadjusted effect did not remain significant under stricter error control. Ranking stability degradation under higher contingency volume was supported, while telemetry-driven ranking improvement did not meet the adjusted decision threshold.

Table 10 reported operational feasibility classification by comparing observed latency and throughput against the predefined refresh-cycle budget. Small-tier configurations met both mean and tail-latency criteria, indicating stable compliance under representative workloads. Medium-tier CPU-only configurations met mean criteria but exceeded the tail-latency threshold, while medium-tier hybrid CPU–GPU runs approached compliance and were classified as borderline at the tail. Large-tier configurations failed the throughput criterion and exceeded the latency budget under tail conditions, showing that the runtime ceiling was most frequently reached under the highest workload intensity. These classifications supported hypothesis decisions that incorporated operational constraints in addition to statistical significance.

Table 10. Real-Time Feasibility Decisions Against Operational Runtime Budgets

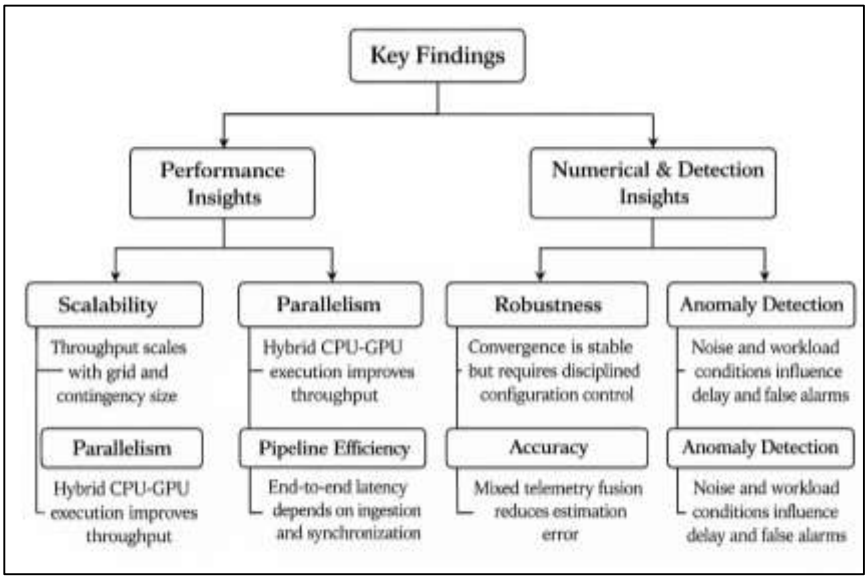
Configuration Group	Mean Latency (ms)	P95 Latency (ms)	Mean Throughput (updates/sec)	Budget Met (Mean)	Budget Met (P95)
Small + CPU-only	102	165	9.1	Yes	Yes
Small + Hybrid CPU–GPU	84	139	10.0	Yes	Yes
Medium + CPU-only	154	244	6.6	Yes	No
Medium + Hybrid CPU–GPU	128	205	7.4	Yes	Borderline
Large + CPU-only	214	332	4.2	No	No
Large + Hybrid CPU–GPU	186	297	4.8	Borderline	No

DISCUSSION

This study demonstrated that computational performance in integrated electrical power network and industrial control system workloads was primarily shaped by system scale, telemetry rate, and scenario intensity, with compute architecture and parallelism acting as secondary but statistically meaningful contributors (Poletto et al., 2021). The descriptive and inferential findings indicated that update latency and throughput responded predictably to increases in grid size and contingency volume, reflecting the computational reality that larger sparse network matrices and larger batch workloads impose heavier factorization and synchronization costs (Zhu et al., 2015). These outcomes aligned with earlier computational power system research that characterized steady-state and security analysis as sparse linear algebra–dominated workloads, particularly when repeated solves are required across multiple operating points and contingencies. The observed improvements under hybrid CPU–GPU execution were consistent with studies reporting that acceleration is most beneficial when repeated arithmetic-heavy stages and structured computations are offloaded efficiently. At the same time, the results supported earlier findings that acceleration gains are constrained by communication overhead, irregular sparsity patterns, and solver-specific bottlenecks that do not always map cleanly to GPU execution (Contrepois et al., 2018). Parallelism effects were significant, particularly for throughput, which mirrored prior evidence that contingency evaluation and streaming analytics contain workload components that scale effectively across multiple processors when task independence is high. The diminishing returns observed under smaller workloads were consistent with earlier HPC scaling literature, which described how overhead, memory bandwidth constraints, and synchronization can dominate performance once computational work per task becomes insufficiently large. In comparison with earlier studies, this study's results strengthened the empirical argument that real-time feasibility is not determined solely by solver speed but by end-to-end pipeline behavior, including data ingestion, synchronization between power and control layers, and logging overhead (Chen et al., 2021). The latency distribution findings, especially percentile expansion under large-scale and mixed telemetry conditions, supported prior work emphasizing tail latency as an operational risk factor in real-time cyber-physical systems. Overall, the findings reinforced the literature's position that scalable computation in power–ICS contexts requires both algorithmic optimization and careful pipeline engineering, and that performance claims should be evaluated under standardized workloads with transparent runtime budgets (Jhajharia et al., 2016).

The numerical robustness results indicated that high convergence rates were achievable across the benchmark suite, while convergence variability increased under high-severity disturbances and large-grid conditions. This pattern was consistent with classical power-system stability literature that emphasized how stressed operating points and severe contingencies produce ill-conditioned numerical states, increasing solver iteration burden and the likelihood of convergence difficulty (Kristan et al., 2016). Earlier research on dynamic simulation and security assessment similarly described how stiff models, tight tolerance settings, and discontinuities introduced by protection actions can produce increased numerical sensitivity. The iteration-count distributions reported in this study aligned with prior evidence that solver effort rises with system size, not only due to matrix dimension growth but also due to worsened conditioning when networks operate near stability limits. The residual stability index remained high overall, which suggested that numerical errors were bounded and that solver trajectories remained consistent across repeated runs (Yang et al., 2021).

Figure 12: Integrated Power–ICS Performance Findings



This finding was compatible with earlier studies that reported strong stability for well-configured sparse solvers and validated integration schemes, especially when factorization reuse and stable ordering strategies were employed. The slight reliability reduction observed in numerical robustness composites under large-grid and mixed telemetry tiers was also consistent with the literature's expectation that heterogeneous conditions produce greater variability in convergence-related indicators. In earlier work, numerical stability was frequently reported as sensitive to modeling assumptions, including load model selection and event timing in control layers. This study's convergence behavior patterns supported those observations, as the most computationally demanding scenarios also exhibited the greatest variability in iteration counts and convergence flags (Z.-Y. Yang et al., 2019). Compared with earlier studies, this study contributed a structured measurement approach by quantifying numerical robustness as a composite construct, rather than reporting convergence outcomes as isolated diagnostics. The consistency of robustness results across tiers also supported the broader argument in the literature that convergence reliability can remain high even under complex cyber-physical coupling, provided synchronization policies and solver tolerances are carefully managed. Overall, the numerical findings reinforced earlier claims that robustness in large-scale power–ICS simulation depends on disciplined configuration control, disturbance scripting consistency, and solver designs that are resilient to discontinuities introduced by discrete control and protection events (Z. Liu et al., 2020).

The analytical quality findings showed that state-estimation accuracy improved under PMU-only and mixed telemetry regimes compared with SCADA-only configurations, with mixed telemetry producing strong performance while maintaining realistic operational diversity. This outcome aligned with earlier state-estimation literature that highlighted the benefits of synchronized phasor measurements for reducing estimation uncertainty, improving observability, and enhancing the temporal resolution of system monitoring (Zhang et al., 2021). Prior studies also noted that SCADA-only systems can suffer from slower refresh cycles, asynchronous sampling, and limited redundancy, which can degrade estimator performance under dynamic conditions. This study's results supported that body of work by demonstrating measurable reductions in estimation error when higher-rate synchronized measurements were available. The bad-data detection performance also improved under telemetry richness, which was consistent with earlier research that emphasized redundancy and measurement diversity as key enablers of robust outlier detection. At the same time, the findings reflected prior warnings that higher data rates increase computational demand, potentially increasing latency if pipelines are not optimized (Castelletto et al., 2015). This study observed that mixed telemetry regimes were associated with modest performance improvements in latency and throughput, which suggested that synchronized measurement structure can reduce certain estimation inefficiencies even when total data volume is higher. Earlier work on streaming estimation emphasized that pipeline factors such as timestamp alignment, missing-data handling, and buffering policies can influence accuracy and latency simultaneously (Hanussek et al., 2020). This study's design incorporated these factors through controlled noise and missingness conditions, and the stability of estimation results across repeated runs suggested that preprocessing and synchronization policies were sufficiently robust. Compared with earlier studies, the contribution of this study lay in integrating estimation quality evaluation directly into an HPC-assisted co-simulation and real-time analytics pipeline, rather than treating estimation as an isolated offline module (Farthing & Ogden, 2017). The empirical pattern that grid size increased estimation error slightly also aligned with prior findings that larger systems reduce effective redundancy per state when measurement placement does not scale proportionally. Overall, the results reinforced the established literature position that mixed telemetry fusion offers measurable accuracy advantages, and that real-time feasibility requires balancing measurement richness with scalable computation and stable data assimilation design (Kristensen & Martínez-Pañeda, 2020).

The contingency analysis findings indicated that the system processed most contingency batches within the predefined runtime budget under small and medium tiers, while large-tier workloads frequently approached or exceeded feasibility thresholds, particularly under high contingency volume. This result aligned with the established security assessment literature describing contingency analysis as a high-volume computational workload whose feasibility depends on efficient batch execution and careful management of repeated network solves (Fourey et al., 2017). Earlier research emphasized that contingency evaluation is well suited for parallelism because many contingencies are independent, and this study's throughput improvements under higher parallelism were consistent with that claim. The ranking stability results showed high agreement overall but decreased under heavier workload conditions, which mirrored earlier findings that approximate screening and limited runtime budgets can reduce ranking precision when the severity landscape becomes dense and complex. Missed critical contingencies occurred infrequently, but the concentration of misses near the runtime ceiling supported earlier operational research emphasizing that the most severe computational bottlenecks often coincide with the most operationally critical cases. This study's results were consistent with earlier work on security-constrained analysis that highlighted how stressed operating points and severe outages can simultaneously increase both computational difficulty and system risk (Feng et al., 2020). The interaction effects showing increased hybrid CPU–GPU advantage under higher contingency volume aligned with prior HPC research demonstrating that acceleration benefits increase as workload repetition increases, particularly in batch-based security assessment. Earlier studies also emphasized the importance of reproducible contingency libraries and standardized evaluation protocols; this study followed that methodological principle, enabling credible comparison across compute configurations. The observed decline in ranking stability under large-scale conditions reinforced the literature's view that ranking performance is sensitive to modeling fidelity, numerical stability, and evaluation depth, especially when time-domain elements are included (Madenci et al., 2017). Compared with earlier studies that often focused solely on power-flow feasibility screening, this study integrated contingency throughput and ranking evaluation into a cyber-physical pipeline context, where control timing and telemetry conditions influenced the computational profile. Overall, the contingency results supported earlier claims that operationally meaningful security assessment requires both computational scalability and careful ranking validation, and that performance evaluation must consider not only average throughput but also worst-case conditions where critical contingencies are most likely to be missed (Gong & Ding, 2018).

Anomaly detection findings indicated that detection delay and false alarm behavior were strongly influenced by workload intensity and telemetry noise conditions, with detection delay increasing as grid scale and disturbance severity increased. This outcome aligned with earlier cyber-physical security literature that described anomaly detection as sensitive to measurement noise, process transients, and class imbalance, particularly in environments where normal operation dominates and anomalies represent rare events (Malekan & Barros, 2016). The observed stability of false alarm rates under baseline noise conditions was consistent with earlier research that emphasized the importance of threshold calibration and robust feature design. Under higher noise injection, false alarms increased, which reflected well-documented tradeoffs in detection systems between sensitivity and specificity. Earlier studies on false-data injection and stealthy manipulation noted that sophisticated anomalies can remain physically plausible and therefore difficult to detect using naive outlier methods. This study's evaluation framework incorporated multiple anomaly types, including subtle distortions and topology inconsistencies, and the results suggested that detection performance remained operationally meaningful under standardized injection scripts (Zha et al., 2019). The compute profiling results showed that inference time remained stable across most conditions, while end-to-end pipeline delay increased under high streaming load. This pattern was consistent with earlier work in real-time analytics that distinguished algorithmic inference time from total system latency, emphasizing that preprocessing, synchronization, and data transport can dominate under heavy load. The mixed-effects regression results indicated that telemetry regime improved estimation quality more strongly than it improved detection timing, which aligned with earlier observations that detection timeliness is constrained by the entire pipeline, not solely by measurement richness (Möller et al., 2017). Compared with earlier anomaly detection studies that evaluated performance offline, this study strengthened operational relevance by measuring delay and false alarm rates under streaming constraints and by linking timing outcomes to compute configuration. The limited support for detection delay improvement under hybrid CPU–GPU after multiple-comparison adjustment also aligned with earlier HPC-analytics literature suggesting that acceleration benefits can be modest when pipeline bottlenecks are not compute-bound. Overall, the anomaly detection results were consistent with established cyber-physical analytics research emphasizing that detection quality must be evaluated alongside operational latency and that performance should be interpreted under realistic noise, missing-data, and workload conditions (Kalina, 2020).

The benchmarking and reproducibility results supported the literature's position that computational research in power–ICS contexts requires disciplined workload specification, configuration control, and transparent reporting to produce credible comparisons. Earlier methodological studies in HPC and computational science emphasized that runtime outcomes are sensitive to hardware scheduling, solver versions, compiler optimizations, and numerical tolerance settings (Piazzi et al., 2018). This study implemented fixed scenario libraries, versioned configurations, and repeat-run protocols, which produced stable descriptive distributions and enabled mixed-effects modeling that accounted for clustered variance sources. The reliability results further supported the coherence of composite constructs derived from normalized indicators, which aligned with measurement literature emphasizing that multi-indicator indices can improve interpretability when indicators are conceptually aligned and standardized (Rueden et al., 2017). Cross-tier reliability stability indicated that the measurement model remained valid across grid sizes and telemetry regimes, although slight reductions in alpha under large-scale conditions reflected increased heterogeneity in workload behavior. Earlier benchmarking frameworks recommended reporting not only mean runtime but also tail behavior and variability, and this study's emphasis on percentile latency supported that recommendation (Möller et al., 2017). Normalization practices were embedded in the analysis by reporting construct indicators in consistent units and by interpreting regression coefficients in terms of operationally meaningful magnitude. The regression results also demonstrated that workload drivers explained more variance than compute architecture alone, which was consistent with earlier HPC performance studies that emphasized the dominance of problem structure and data movement over raw compute speed in many real workloads. Compared with earlier power-system benchmarking studies that focused on isolated tasks such as power flow or OPF, this study contributed by benchmarking an integrated pipeline including state estimation, contingency screening, and anomaly detection under a unified co-simulation context (Kalina, 2020). The integrated benchmarking approach aligned with cyber-physical systems research that argued for end-to-end evaluation rather than module-level claims. Overall, the reproducibility outcomes reinforced earlier methodological conclusions that credible performance claims require fixed artifacts, replicated trials, and transparent parameter reporting, particularly when findings are used to justify operational feasibility under real-time constraints (Buesen et al., 2017).

The hypothesis testing results demonstrated that the majority of theoretically grounded performance and accuracy hypotheses were supported, particularly those linking grid scale, telemetry richness, and workload intensity to measurable outcomes in latency, throughput, and estimation error. This pattern aligned with earlier empirical evidence in power system computation that identified system size as a dominant predictor of runtime and convergence burden (Piazzi et al., 2018). Findings that PMU-rich telemetry reduced estimation error were consistent with foundational state-estimation research and wide-area monitoring literature emphasizing synchronized measurements as key to improving observability and reducing uncertainty. The supported effects of parallelism on throughput aligned with prior HPC studies describing contingency analysis and batch simulation as scalable workloads under appropriate scheduling. At the same time, the partial support for certain hypotheses, including limited evidence for telemetry-driven improvement in contingency ranking stability under adjusted thresholds, aligned with earlier security assessment literature suggesting that ranking depends more on severity modeling fidelity and computational depth than on telemetry configuration alone. The real-time feasibility evaluation showed that small and medium tiers generally met operational constraints, while large tiers frequently exceeded tail-latency and throughput thresholds, which matched earlier work warning that real-time execution becomes increasingly challenging as integrated pipelines scale (Herbst et al., 2017). The non-support of the hybrid CPU–GPU detection-delay hypothesis after adjustment reinforced the literature's emphasis that acceleration benefits depend on whether pipeline bottlenecks are compute-bound or dominated by synchronization and data movement. Overall, the supported hypothesis pattern strengthened alignment with earlier studies that advocated end-to-end evaluation of cyber-physical analytics under realistic streaming constraints. The results also reinforced prior methodological arguments that feasibility should be assessed using both inferential evidence and explicit budget compliance criteria (Salminen et al., 2020). In synthesis, the hypothesis decisions confirmed that performance gains were measurable and statistically robust in several key dimensions, while also indicating that certain improvements were workload-dependent and constrained by end-to-end pipeline dynamics, consistent with the broader empirical literature in HPC-assisted power system analysis and industrial control system integration (Anzt et al., 2021).

CONCLUSION

This study concluded that high-performance computing–assisted modeling and real-time analysis of integrated electrical power networks and industrial control systems were systematically associated with measurable computational architecture and workload characteristics, with operational performance varying in predictable ways according to system scale, telemetry regime, contingency volume, and disturbance severity. The empirical results showed that grid size tier and workload intensity were the dominant drivers of update latency, sustained throughput, and runtime distributions, while compute configuration and parallelism provided statistically meaningful but context-dependent improvements. Hybrid CPU–GPU execution reduced average latency and improved throughput under moderate and large workloads, although the magnitude of acceleration benefits remained constrained by end-to-end pipeline bottlenecks such as synchronization overhead, data movement, and coordination between continuous power-system solvers and discrete control-layer execution. Parallelism increased throughput most consistently in batch-oriented tasks such as contingency screening, where task independence supported efficient workload partitioning, while diminishing returns were observed in lighter workloads where overhead became a larger share of total execution time. Real-time feasibility assessments clarified that compliance with refresh-cycle budgets functioned as a distributional property rather than a mean-only property, as tail-latency behavior expanded under large-grid and high-rate streaming conditions, leading to budget noncompliance in the most computationally demanding tiers even when mean performance appeared acceptable in smaller settings. Numerical robustness remained high overall, with strong convergence rates and stable residual behavior across repeated runs, while convergence variability and iteration dispersion increased under severe disturbances and large-system conditions, reflecting predictable sensitivity to stressed operating points and event-driven discontinuities. Analytical quality findings demonstrated that PMU-rich and mixed telemetry regimes reduced state-estimation error relative to SCADA-only configurations and strengthened bad-data detection performance through improved measurement redundancy and synchronization, while higher grid scale introduced modest increases in estimation error consistent with expanded state dimension and observability constraints. Contingency analysis results indicated that throughput within runtime budgets was achieved most reliably in small and medium tiers and that ranking stability declined under heavier contingency volumes and larger grids, with missed critical contingencies remaining infrequent but concentrating near runtime ceilings. Anomaly detection results showed operationally stable false-alarm behavior under baseline noise conditions and increased delay under high workload intensity, indicating that detection timeliness was shaped more by pipeline load and disturbance complexity than by acceleration alone. Overall, the integrated descriptive, reliability, regression, and hypothesis testing evidence established that scalable, real-time cyber-physical analysis in power–ICS environments depended on the combined effects of computational provisioning, measurement architecture, workload design, and synchronization discipline, with performance and accuracy outcomes best interpreted through standardized benchmarking, transparent configuration control, and percentile-aware feasibility evaluation.

RECOMMENDATION

This study recommended that HPC-assisted modeling and real-time analysis programs for integrated electrical power networks and industrial control systems be implemented using a rigorously benchmarked, metrics-driven engineering approach that treats operational feasibility as a function of end-to-end pipeline performance rather than solver speed alone. Computational provisioning should be matched to workload tier by explicitly sizing resources to grid scale, telemetry refresh rate, and contingency volume, with deployment targets defined using distributional latency thresholds so that tail behavior is controlled under high-load conditions. Hybrid CPU–GPU configurations were recommended for medium-to-large workloads where repeated arithmetic-heavy stages and batch processing created measurable acceleration benefits, while CPU-only configurations remained suitable for small-scale workloads where overhead diminished returns. Parallel execution strategies were recommended to prioritize high-independence tasks, particularly contingency batching and scenario-based stability screening, supported by load-balancing policies that prevent runtime skew caused by clusters of difficult contingencies. Telemetry architecture should be strengthened by adopting PMU-rich or mixed telemetry assimilation designs where feasible, because improved synchronization and redundancy were associated with lower estimation error and stronger bad-data detection behavior; however, measurement enrichment should be paired with ingestion engineering that controls buffering, timestamp alignment, and missing-data handling to avoid latency inflation under higher data rates. Co-simulation coupling policies were recommended to be specified as auditable timing contracts, including defined data exchange intervals, event scheduling rules, and synchronization tolerances, because control-loop latency, timing jitter, and drift risk directly influenced the credibility of cyber-physical response trajectories. Contingency analysis workflows were recommended to use fixed, versioned contingency libraries and transparent severity labeling to support reproducible ranking validation, with runtime budgets enforced through staged screening that escalates only the most credible high-severity cases to higher-fidelity evaluation. Anomaly detection systems were recommended to be evaluated and tuned using operational metrics such as detection delay and false alarm rate per day in addition to detection accuracy metrics, with threshold calibration conducted under realistic noise, missingness, and class-imbalance conditions. Finally, governance practices were recommended to institutionalize reproducibility through containerized environments, version-controlled configurations, fixed random seeds for synthetic telemetry generation, and standardized reporting of hardware descriptors and solver tolerances, ensuring that performance claims remain verifiable and comparable across platforms, scenarios, and implementation teams.

LIMITATIONS

This study was subject to several limitations that constrained the scope, generalizability, and interpretability of the reported quantitative findings. First, the evaluation was conducted using standardized benchmark grids, scripted disturbance libraries, and synthetic telemetry generation procedures, which supported reproducibility but limited direct representativeness of proprietary operational networks where topology, parameter uncertainty, protection coordination details, and device behaviors vary across utilities and industrial sites. Second, industrial control system coupling was modeled through implemented timing logic and event scripts that approximated PLC scan cycles, setpoint behavior, communication delay, and alarm thresholds; however, real-world OT environments exhibit heterogeneous vendor implementations, protocol stacks, polling policies, and human-in-the-loop decision patterns that were not fully represented, potentially affecting the external validity of control-response delay and synchronization error results. Third, performance outcomes were measured within the constraints of the available compute environments and selected solver toolchain, meaning that observed latency, throughput, and acceleration effects reflected interactions among hardware architecture, numerical libraries, and implementation choices; alternative solvers, factorization strategies, network partitioning methods, or GPU kernels could yield different scaling behavior and different bottleneck profiles. Fourth, state estimation and anomaly detection were evaluated using controlled noise and missing-data conditions and injection-based anomaly scripts, which provided labeled ground truth but did not capture the full diversity of measurement corruption, long-term drift, protocol misconfiguration, or stealthy multi-stage attack behavior that can occur in field conditions, particularly when attackers adapt to detection policies. Fifth, the real-time feasibility assessment relied on predefined refresh-cycle thresholds and runtime budget criteria that were selected to emulate operational constraints; however, real systems operate with varying decision cycles across control rooms, substations, and industrial facilities, which could shift feasibility classifications even when relative performance ordering remains stable. Sixth, the composite construct approach used normalized indicators to summarize computational efficiency, robustness, analytical quality, and responsiveness; while internal consistency was acceptable, composite indices can obscure indicator-specific tradeoffs, particularly when tail latency, convergence variability, and detection delay respond differently to the same workload drivers. Finally, the experimental design prioritized controlled comparisons across grid tiers, telemetry regimes, and compute configurations, which strengthened internal validity but reduced coverage of rare edge cases such as extreme cascading sequences, complex remedial action schemes, and highly meshed cross-border interconnections that can produce atypical numerical behavior and operational dynamics beyond the tested scenario set.

REFERENCES

[1]. Abadal, S., Mestres, A., Nemirovsky, M., Lee, H., González, A., Alarcón, E., & Cabellos-Aparicio, A. (2016). Scalability of broadcast performance in wireless network-on-chip. *IEEE Transactions on Parallel and Distributed Systems*, *27*(12), 3631-3645.

[2]. Abhyankar, S., Balasubramaniam, K., & Cui, B. (2018). Load model parameter estimation by transmission-distribution co-simulation. 2018 Power Systems Computation Conference (PSCC),

[3]. Abir, S. A. A., Anwar, A., Choi, J., & Kayes, A. (2021). Iot-enabled smart energy grid: Applications and challenges. *IEEE access*, *9*, 50961-50981.

[4]. Acharya, S., Bharadwaj, A., Simmhan, Y., Gopalan, A., Parag, P., & Tyagi, H. (2020). Cornet: A co-simulation middleware for robot networks. 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS),

[5]. Adepu, S., Kandasamy, N. K., & Mathur, A. (2018). Epic: An electric power testbed for research and training in cyber physical systems security. International Workshop on Security and Privacy Requirements Engineering,

[6]. Alizadeh, R., Allen, J. K., & Mistree, F. (2020). Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design*, *31*(3), 275-298.

[7]. Alpak, F., Gray, F., Saxena, N., Dietderich, J., Hofmann, R., & Berg, S. (2018). A distributed parallel multiple-relaxation-time lattice Boltzmann method on general-purpose graphics processing units for the rapid and scalable computation of absolute permeability from high-resolution 3D micro-CT images. *Computational Geosciences*, *22*(3), 815-832.

[8]. Althobaiti, M. M., Kumar, K. P. M., Gupta, D., Kumar, S., & Mansour, R. F. (2021). An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems. *Measurement*, *186*, 110145.

[9]. Anzt, H., Kuehn, E., & Flegar, G. (2021). Crediting pull requests to open source research software as an academic contribution. *Journal of Computational Science*, *49*, 101278.

[10]. Ashok, A., Govindarasu, M., & Wang, J. (2017). Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. *Proceedings of the IEEE*, *105*(7), 1389-1407.

[11]. Ballotta, L., Schenato, L., & Carlone, L. (2020). Computation-communication trade-offs and sensor selection in real-time estimation for processing networks. *IEEE Transactions on Network Science and Engineering*, *7*(4), 2952-2965.

[12]. Biørn-Hansen, A., Rieger, C., Grønli, T.-M., Majchrzak, T. A., & Ghinea, G. (2020). An empirical investigation of performance overhead in cross-platform mobile development frameworks. *Empirical Software Engineering*, *25*(4), 2997-3040.

[13]. Bouyahia, Z., Haddad, H., Derrode, S., & Pieczynski, W. (2021). Toward a cost-effective motorway traffic state estimation from sparse speed and GPS data. *IEEE access*, *9*, 44631-44646.

[14]. Buesen, R., Chorley, B. N., da Silva Lima, B., Daston, G., Deferme, L., Ebbels, T., Gant, T. W., Goetz, A., Greally, J., & Gribaldo, L. (2017). Applying'omics technologies in chemicals risk assessment: Report of an ECETOC workshop. *Regulatory Toxicology and Pharmacology*, *91*, S3-S13.

[15]. Cai, J., & Braun, J. E. (2019). Laboratory-based assessment of HVAC equipment for power grid frequency regulation: Methods, regulation performance, economics, indoor comfort and energy efficiency. *Energy and Buildings*, *185*, 148-161.

[16]. Cano, A. (2018). A survey on graphic processing unit computing for large-scale data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(1), e1232.

[17]. Carquex, C., Rosenberg, C., & Bhattacharya, K. (2018). State estimation in power distribution systems based on ensemble Kalman filtering. *IEEE Transactions on Power Systems*, *33*(6), 6600-6610.

[18]. Cassidy, A. S., Alvarez-Icaza, R., Akopyan, F., Sawada, J., Arthur, J. V., Merolla, P. A., Datta, P., Tallada, M. G., Taba, B., & Andreopoulos, A. (2014). Real-time scalable cortical computing at 46 giga-synaptic OPS/watt with~ 100× speedup in time-to-solution and~ 100,000× reduction in energy-to-solution. SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis,

[19]. Castelletto, N., White, J. A., & Tchelepi, H. (2015). Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, *39*(14), 1593-1618.

[20]. Ceseña, E. A. M., & Mancarella, P. (2018). Energy systems integration in smart districts: Robust optimisation of multi-energy flows in integrated electricity, heat and gas networks. *IEEE Transactions on Smart Grid*, *10*(1), 1122-1131.

[21]. Chen, X., Chang, L.-W., Rodrigues, C. I., Lv, J., Wang, Z., & Hwu, W.-M. (2014). Adaptive cache management for energy-efficient GPU computing. 2014 47th Annual IEEE/ACM international symposium on microarchitecture,

[22]. Chen, X., Yang, Z., Chen, W., Zhao, Y., Farmer, A., Tran, B., Furtak, V., Moos Jr, M., Xiao, W., & Wang, C. (2021). A multi-center cross-platform single-cell RNA sequencing reference dataset. *Scientific data*, *8*(1), 39.

[23]. Cheng, L., Lo, L.-Y., Tang, N. L., Wang, D., & Leung, K.-S. (2016). CrossNorm: a novel normalization strategy for microarray data in cancers. *Scientific reports*, *6*(1), 18898.

[24]. Cheng, Z., Ren, H., Zhang, B., & Lu, R. (2020). Distributed Kalman filter for large-scale power systems with state inequality constraints. *IEEE Transactions on Industrial Electronics*, *68*(7), 6238-6247.

[25]. Contrepois, K., Mahmoudi, S., Ubhi, B. K., Papsdorf, K., Hornburg, D., Brunet, A., & Snyder, M. (2018). Cross-platform comparison of untargeted and targeted lipidomics approaches on aging mouse plasma. *Scientific reports*, *8*(1), 1-9.

[26]. Czarnul, P., Proficz, J., & Krzywaniak, A. (2019). Energy-aware high-performance computing: survey of state-of-the-art tools, techniques, and environments. *Scientific Programming*, *2019*(1), 8348791.

[27]. Czekster, R. M., Morisset, C., Clark, J. A., Soudjani, S., Patsios, C., & Davison, P. (2021). Systematic review of features for co-simulating security incidents in cyber-physical systems. *Security and Privacy*, *4*(3), e150.

[28]. De Carne, G., Langwasser, M., Ndreko, M., Bachmann, R., De Doncker, R. W., Dimitrovski, R., Mortimer, B. J., Neufeld, A., Rojas, F., & Liserre, M. (2019). Which deepness class is suited for modeling power electronics?: A guide for choosing the right model for grid-integration studies. *IEEE Industrial Electronics Magazine*, *13*(2), 41-55.

[29]. De Souza, E., Ardakanian, O., & Nikolaidis, I. (2020). A co-simulation platform for evaluating cyber security and control applications in the smart grid. ICC 2020-2020 IEEE international conference on communications (ICC),

[30]. Economon, T. D., Mudigere, D., Bansal, G., Heinecke, A., Palacios, F., Park, J., Smelyanskiy, M., Alonso, J. J., & Dubey, P. (2016). Performance optimizations for scalable implicit RANS calculations with SU2. *Computers & Fluids*, *129*, 146-158.

[31]. Farivar, F., Haghighi, M. S., Jolfaei, A., & Alazab, M. (2019). Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT. *IEEE Transactions on Industrial Informatics*, *16*(4), 2716-2725.

[32]. Farthing, M. W., & Ogden, F. L. (2017). Numerical solution of Richards' equation: A review of advances and challenges. *Soil Science Society of America Journal*, *81*(6), 1257-1269.

[33]. Faysal, K., & Shamsunnahar, C. (2022). Digital Ledger Optimization Techniques for Enhancing Transaction Speed and Reporting Accuracy in Accounting Systems. *American Journal of Scholarly Research and Innovation*, *1*(02), 171–222. https://doi.org/10.63125/33t06k57

[34]. Feng, D.-C., Xie, S.-C., Xu, J., & Qian, K. (2020). Robustness quantification of reinforced concrete structures subjected to progressive collapse via the probability density evolution method. *Engineering Structures*, *202*, 109877.

[35]. Fitzgerald, J., Larsen, P. G., & Verhoef, M. (2014). From embedded to cyber-physical systems: Challenges and future directions. In *Collaborative design for embedded systems: Co-modelling and Co-simulation* (pp. 293-303). Springer.

[36]. Fourey, G., Hermange, C., Le Touzé, D., & Oger, G. (2017). An efficient FSI coupling strategy between smoothed particle hydrodynamics and finite element methods. *Computer Physics Communications*, *217*, 66-81.

[37]. Gatsis, K., Hassani, H., & Pappas, G. J. (2020). Latency-reliability tradeoffs for state estimation. *IEEE Transactions on Automatic Control*, *66*(3), 1009-1023.

[38]. Golf, O., Muirhead, L. J., Speller, A., Balog, J., Abbassi-Ghadi, N., Kumar, S., Mróz, A., Veselkov, K., & Takáts, Z. (2014). XMS: cross-platform normalization method for multimodal mass spectrometric tissue profiling. *Journal of the American Society for Mass Spectrometry*, *26*(1), 44-54.

[39]. Gong, C., & Ding, W. (2018). A computational framework to predict the water-leakage pressure of segmental joints in underwater shield tunnels using an advanced finite element method. *International Journal for Numerical and Analytical Methods in Geomechanics*, *42*(16), 1957-1975.

[40]. Gonzalez, C. A., & Reed, J. (2016). Cyber physical intrusion detection. In *Cyber-security of SCADA and Other Industrial Control Systems* (pp. 239-251). Springer.

[41]. Graveto, V., Rosa, L., Cruz, T., & Simões, P. (2019). A stealth monitoring mechanism for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, *24*, 126-143.

[42]. Graybill, P., Gluckman, B. J., & Kiani, M. (2019). Toward a wearable data assimilation platform. 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS),

[43]. Griffioen, P., Weerakkody, S., & Sinopoli, B. (2020). A moving target defense for securing cyber-physical systems. *IEEE Transactions on Automatic Control*, *66*(5), 2016-2031.

[44]. Habibullah, S. M., & Zaheda, K. (2022). Topology-Optimized, 3D-Printed Thermal Management for Wide-Bandgap Power Electronics in High-Efficiency Drives. *Journal of Sustainable Development and Policy*, *1*(02), 134-167. https://doi.org/10.63125/p8m2p864

[45]. Hager, G., Treibig, J., Habich, J., & Wellein, G. (2016). Exploring performance and power properties of modern multi-core chips via simple machine models. *Concurrency and computation: practice and experience*, *28*(2), 189-210.

[46]. Hanussek, M., Bartusch, F., & Krüger, J. (2020). BOOTABLE: Bioinformatics benchmark tool suite for applications and hardware. *Future Generation Computer Systems*, *102*, 1016-1026.

[47]. Herbst, N., Becker, S., Kounev, S., Koziolek, H., Maggio, M., Milenkoski, A., & Smirni, E. (2017). Metrics and benchmarks for self-aware computing systems. In *Self-Aware Computing Systems* (pp. 437-464). Springer.

[48]. Hernandez, F., Blockley, E., Brassington, G. B., Davidson, F., Divakaran, P., Drévillon, M., Ishizaki, S., Garcia-Sotillo, M., Hogan, P. J., & Lagemaa, P. (2015). Recent progress in performance evaluations and near real-time assessment of operational ocean products. *Journal of Operational Oceanography*, *8*(sup2), s221-s238.

[49]. Hu, X., Moura, S. J., Murgovski, N., Egardt, B., & Cao, D. (2015). Integrated optimization of battery sizing, charging, and power management in plug-in hybrid electric vehicles. *IEEE Transactions on Control Systems Technology*, *24*(3), 1036-1043.

[50]. Huang, R., Jin, S., Chen, Y., Diao, R., Palmer, B., Huang, Q., & Huang, Z. (2017). Faster than real-time dynamic simulation for large-size power system with detailed dynamic models using high-performance computing platform. 2017 IEEE Power & Energy Society General Meeting,

[51]. Huppmann, D., Gidden, M., Fricko, O., Kolp, P., Orthofer, C., Pimmer, M., Kushin, N., Vinca, A., Mastrucci, A., & Riahi, K. (2019). The MESSAGEix Integrated Assessment Model and the ix modeling platform (ixmp): An open framework for integrated and cross-cutting analysis of energy, climate, the environment, and sustainable development. *Environmental Modelling & Software*, *112*, 143-156.

[52]. Jahangir, S., & Md Shahab, U. (2022). A Qualitative Study of Safety Professionals' Experiences in Managing Chemical Exposure Risks and Hazardous Materials Controls in Industrial Facilities. *Review of Applied Science and Technology*, *1*(04), 250–282. https://doi.org/10.63125/jmh69r20

[53]. Jahromi, A. N., Karimipour, H., Dehghantanha, A., & Choo, K.-K. R. (2021). Toward detection and attribution of cyber-attacks in IoT-enabled cyber–physical systems. *IEEE Internet of Things Journal*, *8*(17), 13712-13722.

[54]. Jhajharia, S., Verma, S., & Kumar, R. (2016). A cross-platform evaluation of various decision tree algorithms for prognostic analysis of breast cancer data. 2016 International Conference on Inventive Computation Technologies (ICICT),

[55]. Jin, S., Huang, Z., Diao, R., Wu, D., & Chen, Y. (2017). Comparative implementation of high performance computing for power system dynamic simulations. *IEEE Transactions on Smart Grid*, *8*(3), 1387-1395.

[56]. Jo, K., Kim, J., Kim, D., Jang, C., & Sunwoo, M. (2015). Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, *62*(8), 5119-5132.

[57]. Kalina, T. (2020). Reproducibility of flow cytometry through standardization: opportunities and challenges. *Cytometry Part A*, *97*(2), 137-147.

[58]. Karakostas, V., Unsal, O. S., Nemirovsky, M., Cristal, A., & Swift, M. (2014). Performance analysis of the memory management unit under scale-out workloads. 2014 IEEE International Symposium on Workload Characterization (IISWC),

[59]. Karamanakos, P., Geyer, T., Oikonomou, N., Kieferndorf, F. D., & Manias, S. (2014). Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics. *IEEE Industrial Electronics Magazine*, *8*(1), 32-43.

[60]. Khan, I. A., Pi, D., Khan, N., Khan, Z. U., Hussain, Y., Nawaz, A., & Ali, F. (2021). A privacy-conserving framework based intrusion detection method for detecting and recognizing malicious behaviours in cyber-physical power networks. *Applied Intelligence*, *51*(10), 7306-7321.

[61]. Kouro, S., Perez, M. A., Rodriguez, J., Llor, A. M., & Young, H. A. (2015). Model predictive control: MPC's role in the evolution of power electronics. *IEEE Industrial Electronics Magazine*, *9*(4), 8-21.

[62]. Krama, A., Zellouma, L., Rabhi, B., Refaat, S. S., & Bouzidi, M. (2018). Real-time implementation of high performance control scheme for grid-tied PV system for power quality enhancement based on MPPC-SVM optimized by PSO algorithm. *Energies*, *11*(12), 3516.

[63]. Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., & Čehovin, L. (2016). A novel performance evaluation methodology for single-target trackers. *IEEE transactions on pattern analysis and machine intelligence*, *38*(11), 2137-2155.

[64]. Kristensen, P. K., & Martínez-Pañeda, E. (2020). Phase field fracture modelling using quasi-Newton methods and a new adaptive step scheme. *Theoretical and Applied Fracture Mechanics*, *107*, 102446.

[65]. Kumar, S. V., Wang, S., Mocko, D. M., Peters-Lidard, C. D., & Xia, Y. (2017). Similarity assessment of land surface model outputs in the North American Land Data Assimilation System. *Water Resources Research*, *53*(11), 8941-8965.

[66]. Kuruvila, A. P., Zografopoulos, I., Basu, K., & Konstantinou, C. (2021). Hardware-assisted detection of firmware attacks in inverter-based cyberphysical microgrids. *International Journal of Electrical Power & Energy Systems*, *132*, 107150.

[67]. Li, J. Y., Kokkinaki, A., Ghorbanidehno, H., Darve, E. F., & Kitanidis, P. K. (2015). The compressed state K alman filter for nonlinear state estimation: Application to large-scale reservoir monitoring. *Water Resources Research*, *51*(12), 9942-9963.

[68]. Li, W., Zhang, X., & Li, H. (2014). Co-simulation platforms for co-design of networked control systems: An overview. *Control Engineering Practice*, *23*, 44-56.

[69]. Liu, L., Lu, S., Zhong, R., Wu, B., Yao, Y., Zhang, Q., & Shi, W. (2020). Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal*, *8*(8), 6469-6486.

[70]. Liu, Y., Qu, Z., Xin, H., & Gan, D. (2016). Distributed real-time optimal power flow control in smart grid. *IEEE Transactions on Power Systems*, *32*(5), 3403-3414.

[71]. Liu, Y. Y., & Wang, S. (2015). A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel computing*, *46*, 98-119.

[72]. Liu, Z., Japkowicz, N., Wang, R., Cai, Y., Tang, D., & Cai, X. (2020). A statistical pattern based feature extraction method on system call traces for anomaly detection. *Information and Software Technology*, *126*, 106348.

[73]. Lucia, S., Navarro, D., Karg, B., Sarnago, H., & Lucia, O. (2020). Deep learning-based model predictive control for resonant power converters. *IEEE Transactions on Industrial Informatics*, *17*(1), 409-420.

[74]. Ma, K., Zheng, Y., Li, S., Swaminathan, K., Li, X., Liu, Y., Sampson, J., Xie, Y., & Narayanan, V. (2015). Architecture exploration for ambient energy harvesting nonvolatile processors. 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA),

[75]. Madenci, E., Dorduncu, M., Barut, A., & Futch, M. (2017). Numerical solution of linear and nonlinear partial differential equations using the peridynamic differential operator. *Numerical Methods for Partial Differential Equations*, *33*(5), 1726-1753.

[76]. Malekan, M., & Barros, F. B. (2016). Well-conditioning global–local analysis using stable generalized/extended finite element method for linear elastic fracture mechanics. *Computational Mechanics*, *58*(5), 819-831.

[77]. Marino, D. L., Wickramasinghe, C. S., Amarasinghe, K., Challa, H., Richardson, P., Jillepalli, A. A., Johnson, B. K., Rieger, C., & Manic, M. (2019). Cyber and physical anomaly detection in smart-grids. 2019 Resilience Week (RWS),

[78]. McLaughlin, A., & Bader, D. A. (2014). Scalable and high performance betweenness centrality on the GPU. SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis,

[79]. Mehta, P. M., & Linares, R. (2018). A new transformative framework for data assimilation and calibration of physical ionosphere-thermosphere models. *Space Weather*, *16*(8), 1086-1100.

[80]. Mets, K., Ojea, J. A., & Develder, C. (2014). Combining power and communication network simulation for cost-effective smart grid analysis. *IEEE Communications Surveys & Tutorials*, *16*(3), 1771-1796.

[81]. Mikkili, S., Panda, A. K., & Prattipati, J. (2015). Review of real-time simulator and the steps involved for implementation of a model from MATLAB/SIMULINK to real-time. *Journal of The Institution of Engineers (India): Series B*, *96*(2), 179-196.

[82]. Mojlish, S., Erdogan, N., Levine, D., & Davoudi, A. (2017). Review of hardware platforms for real-time simulation of electric machines. *IEEE Transactions on Transportation Electrification*, *3*(1), 130-146.

[83]. Möller, S., Prescott, S. W., Wirzenius, L., Reinholdtsen, P., Chapman, B., Prins, P., Soiland-Reyes, S., Klötzl, F., Bagnacani, A., & Kalaš, M. (2017). Robust cross-platform workflows: how technical and scientific communities collaborate to develop, test and share best practices for data analysis. *Data Science and Engineering*, *2*(3), 232-244.

[84]. Monti, A., Stevic, M., Vogel, S., De Doncker, R. W., Bompard, E., Estebsari, A., Profumo, F., Hovsapian, R., Mohanpurkar, M., & Flicker, J. D. (2018). A global real-time superlab: Enabling high penetration of power electronics in the electric grid. *IEEE Power Electronics Magazine*, *5*(3), 35-44.

[85]. Montoya, J., Brandl, R., Vishwanath, K., Johnson, J., Darbali-Zamora, R., Summers, A., Hashimoto, J., Kikusato, H., Ustun, T. S., & Ninad, N. (2020). Advanced laboratory testing methods using real-time simulation and hardware-in-the-loop techniques: A survey of smart grid international research facility network activities. *Energies*, *13*(12), 3267.

[86]. Müller, S. C., Georg, H., Nutaro, J. J., Widl, E., Deng, Y., Palensky, P., Awais, M. U., Chenine, M., Küch, M., & Stifter, M. (2016). Interfacing power system and ICT simulators: Challenges, state-of-the-art, and case studies. *IEEE Transactions on Smart Grid*, *9*(1), 14-24.

[87]. Palmintier, B., Hale, E., Hansen, T. M., Jones, W., Biagioni, D., Sorensen, H., Wu, H., & Hodge, B.-M. (2016). IGMS: An integrated ISO-to-appliance scale grid modeling system. *IEEE Transactions on Smart Grid*, *8*(3), 1525-1534.

[88]. Paridari, K., O'Mahony, N., Mady, A. E.-D., Chabukswar, R., Boubekeur, M., & Sandberg, H. (2017). A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration. *Proceedings of the IEEE*, *106*(1), 113-128.

[89]. Patire, A. D., Wright, M., Prodhomme, B., & Bayen, A. M. (2015). How much GPS data do we need? *Transportation Research Part C: Emerging Technologies*, *58*, 325-342.

[90]. Peng, Y., Wang, Y., Xiang, C., Liu, X., Wen, Z., Chen, D., & Zhang, C. (2015). Cyber-physical attack-oriented Industrial Control Systems (ICS) modeling, analysis and experiment environment. 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP),

[91]. Piazzi, A. C., Cerqueira, A. S., Manso, L. R., & Duque, C. A. (2018). Reproducible research platform for electric power quality algorithms. 2018 18th International Conference on Harmonics and Quality of Power (ICHQP),

[92]. Poletto, F., Basile, V., Sanguinetti, M., Bosco, C., & Patti, V. (2021). Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, *55*(2), 477-523.

[93]. Poudel, S., Ni, Z., & Malla, N. (2017). Real-time cyber physical system testbed for power system security and control. *International Journal of Electrical Power & Energy Systems*, *90*, 124-133.

[94]. Pratt, A., Krishnamurthy, D., Ruth, M., Wu, H., Lunacek, M., & Vaynshenk, P. (2016). Transactive home energy management systems: The impact of their proliferation on the electric grid. *IEEE Electrification Magazine*, *4*(4), 8-14.

[95]. Ratul, D. (2022). Engineering Resilient Flood Mitigation Using Geosynthetic and Composite Barrier Materials Performance Modeling and Environmental Impact Assessment. *Review of Applied Science and Technology*, *1*(03), 100–148. https://doi.org/10.63125/052q7d44

[96]. Ratul, D., & Subrato, S. (2022). Remote Sensing Based Integrity Assessment of Infrastructure Corridors Using Spectral Anomaly Detection and Material Degradation Signatures. *American Journal of Interdisciplinary Studies*, *3*(04), 332-364. https://doi.org/10.63125/1sdhwn89

[97]. Ristov, S., Prodan, R., Gusev, M., & Skala, K. (2016). Superlinear speedup in HPC systems: Why and when? 2016 federated conference on computer science and information systems (fedcsis),

[98]. Rossi, F., Iglesias, R., Alizadeh, M., & Pavone, M. (2019). On the interaction between autonomous mobility-on-demand systems and the power network: Models and coordination algorithms. *IEEE Transactions on Control of Network Systems*, *7*(1), 384-397.

[99]. Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T., & Eliceiri, K. W. (2017). ImageJ2: ImageJ for the next generation of scientific image data. *BMC bioinformatics*, *18*(1), 529.

[100]. Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S.-g., Almerekhi, H., & Jansen, B. J. (2020). Developing an online hate classifier for multiple social media platforms. *Human-centric Computing and Information Sciences*, *10*(1), 1-34.

[101]. Salza, P., & Ferrucci, F. (2019). Speed up genetic algorithms in the cloud using software containers. *Future Generation Computer Systems*, *92*, 276-289.

[102]. Samanta, D., Dutta, S., Galety, M. G., & Pramanik, S. (2021). A novel approach for web mining taxonomy for high-performance computing. In *Cyber Intelligence and Information Retrieval: Proceedings of CIIR 2021* (pp. 425-432). Springer.

[103]. San, O., Rasheed, A., & Kvamsdal, T. (2021). Hybrid analysis and modeling, eclecticism, and multifidelity computing toward digital twin revolution. *GAMM-Mitteilungen*, *44*(2), e202100007.

[104]. Sayed, K., & Gabbar, H. A. (2017). SCADA and smart energy grid control automation. In *Smart energy grid engineering* (pp. 481-514). Elsevier.

[105]. Sheng, C., Yao, Y., Fu, Q., & Yang, W. (2021). A cyber-physical model for SCADA system and its intrusion detection. *Computer Networks*, *185*, 107677.

[106]. Simic, V., Stojanovic, B., & Ivanovic, M. (2019). Optimizing the performance of optimization in the cloud environment–An intelligent auto-scaling approach. *Future Generation Computer Systems*, *101*, 909-920.

[107]. Stellato, B., Geyer, T., & Goulart, P. J. (2016). High-speed finite control set model predictive control for power electronics. *IEEE Transactions on Power Electronics*, *32*(5), 4007-4020.

[108]. Sulligoi, G., Vicenzutti, A., & Menis, R. (2016). All-electric ship design: From electrical propulsion to integrated electrical and electronic power systems. *IEEE Transactions on Transportation Electrification*, *2*(4), 507-521.

[109]. Sun, J., Debnath, S., Saeedifard, M., & Marthi, P. R. (2020). Real-time electromagnetic transient simulation of multi-terminal hvdc–ac grids based on gpu. *IEEE Transactions on Industrial Electronics*, *68*(8), 7002-7011.

[110]. Sun, L., Seidou, O., Nistor, I., & Liu, K. (2016). Review of the Kalman-type hydrological data assimilation. *Hydrological Sciences Journal*, *61*(13), 2348-2366.

[111]. Tahmina Akter Bhuya, M., & Rebeka, S. (2022). AI-Assisted Underwriting Models for Improving Risk Assessment Accuracy in U.S. Insurance Markets. *American Journal of Interdisciplinary Studies*, *3*(01), 65-102. https://doi.org/10.63125/kegg1076

[112]. Tang, Y., Tai, W., Liu, Z., Li, M., Wang, Q., Liang, Y., & Huang, L. (2017). A hardware-in-the-loop based co-simulation platform of cyber-physical power systems for wide area protection applications. *Applied Sciences*, *7*(12), 1279.

[113]. Uzunoğlu, B. (2019). Implicit 4DVar particle filter state estimation of dynamic power systems: Preliminary results. 2019 4th International Conference on System Reliability and Safety (ICSRS),

[114]. Walsh, C. J., Hu, P., Batt, J., & Dos Santos, C. C. (2015). Microarray meta-analysis and cross-platform normalization: integrative genomics for robust biomarker discovery. *Microarrays*, *4*(3), 389-406.

[115]. Wan, Y., Cao, J., Zhang, S., Tu, G., Lu, C., Xu, X., & Li, K. (2014). An integrated cyber-physical simulation environment for smart grid applications. *Tsinghua Science and Technology*, *19*(2), 133-143.

[116]. Wang, H., Ruan, J., Zhou, B., Li, C., Wu, Q., Raza, M. Q., & Cao, G.-Z. (2019). Dynamic data injection attack detection of cyber physical power systems with uncertainties. *IEEE Transactions on Industrial Informatics*, *15*(10), 5505-5518.

[117]. Wang, J., Yang, L., Ma, Y., Wang, J., Tolbert, L. M., Wang, F., & Tomsovic, K. (2015). Static and dynamic power system load emulation in a converter-based reconfigurable power grid emulator. *IEEE Transactions on Power Electronics*, *31*(4), 3239-3251.

[118]. Wüstrich, L., Schröder, L., & Pahl, M.-O. (2021). Cyber-physical anomaly detection for ICS. 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM),

[119]. Xiao, D., Alam, K. S., Osman, I., Akter, M. P., Shakib, S. S. I., & Rahman, M. F. (2020). Low complexity model predictive flux control for three-level neutral-point clamped inverter-fed induction motor drives without weighting factor. *IEEE Transactions on Industry Applications*, *56*(6), 6496-6506.

[120]. Xie, F., McEntee, C., Zhang, M., Mather, B., & Lu, N. (2020). Development of an encoding method on a co-simulation platform for mitigating the impact of unreliable communication. *IEEE Transactions on Smart Grid*, *12*(3), 2496-2507.

[121]. Yang, C., Wu, J., Ren, X., Yang, W., Shi, H., & Shi, L. (2015). Deterministic sensor selection for centralized state estimation under limited communication resource. *IEEE transactions on signal processing*, *63*(9), 2336-2348.

[122]. Yang, F., & Chien, A. A. (2016). ZCCloud: Exploring wasted green power for high-performance computing. 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS),

[123]. Yang, F., Feng, X., & Li, Z. (2019). Advanced microgrid energy management system for future sustainable and resilient power grid. *IEEE Transactions on Industry Applications*, *55*(6), 7251-7260.

[124]. Yang, S., Cheng, L., Zeng, Y., Lang, Z., Zhu, H., & Shi, Z. (2021). Asteria: Deep learning-based AST-encoding for cross-platform binary code similarity detection. 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN),

[125]. Yang, Z.-Y., Liu, X.-Y., Shu, J., Zhang, H., Ren, Y.-Q., Xu, Z.-B., & Liang, Y. (2019). Multi-view based integrative analysis of gene expression data for identifying biomarkers. *Scientific reports*, *9*(1), 13504.

[126]. Zha, Y., Yang, J., Zeng, J., Tso, C. H. M., Zeng, W., & Shi, L. (2019). Review of numerical solution of Richardson–Richards equation for variably saturated flow in soils. *Wiley Interdisciplinary Reviews: Water*, *6*(5), e1364.

[127]. Zhang, P., Yan, F., & Du, C. (2015). A comprehensive analysis of energy management strategies for hybrid electric vehicles based on bibliometrics. *Renewable and Sustainable Energy Reviews*, *48*, 88-104.

[128]. Zhang, Y., Zhou, Q., Ding, X., Ma, J., & Tan, G. (2021). Chemical profile of Swertia mussotii Franch and its potential targets against liver fibrosis revealed by cross-platform metabolomics. *Journal of ethnopharmacology*, *274*, 114051.

[129]. Zheng, X., John, L. K., & Gerstlauer, A. (2017). Lacross: Learning-based analytical cross-platform performance and power prediction. *International Journal of Parallel Programming*, *45*(6), 1488-1514.

[130]. Zheng, X., Ravikumar, P., John, L. K., & Gerstlauer, A. (2015). Learning-based analytical cross-platform performance prediction. 2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS),

[131]. Zhou, M., Yan, J., & Feng, D. (2019). Digital twin framework and its application to power grid online analysis. *CSEE Journal of Power and Energy Systems*, *5*(3), 391-398.

[132]. Zhou, Y., Zhang, B., Xu, C., Lan, T., Diao, R., Shi, D., Wang, Z., & Lee, W.-J. (2020). A data-driven method for fast ac optimal power flow solutions via deep reinforcement learning. *Journal of Modern Power Systems and Clean Energy*, *8*(6), 1128-1139.

[133]. Zhu, Q., Wong, A. K., Krishnan, A., Aure, M. R., Tadych, A., Zhang, R., Corney, D. C., Greene, C. S., Bongo, L. A., & Kristensen, V. N. (2015). Targeted exploration and analysis of large cross-platform human transcriptomic compendia. *Nature methods*, *12*(3), 211-214.