pISSN:2349-5707 Volume:12, Issue 07, JULY-2025 eISSN:2349-5715

Resilient, Automated Monitoring and Fault-Tolerant Control for Critical Building Systems: Integrating GPU-Accelerated Anomaly Detection, Infrastructure-as-Code, and Self-Correcting HVAC Strategies

Dr. Elena Moretti
Institute for Systems Resilience, University of Lausanne

ABSTRACT

This article presents an integrative framework for designing resilient, automated monitoring and fault-tolerant control systems for critical building infrastructure, with an emphasis on heating, ventilation, and air-conditioning (HVAC) systems. The proposed framework synthesizes advances in GPU-accelerated anomaly detection, infrastructure-as-code (IaC) for reproducible deployment, self-correcting control strategies, and organizational preparedness in emergency scenarios. The study draws on interdisciplinary evidence spanning disaster and emergency planning, GPU concurrency and diagnostic automation, advanced anomaly detection in large sensor networks, automated deployment methods, and domain-specific best practices for HVAC control and fault detection. The framework is organized into four interlocking pillars: (1) high-throughput real-time anomaly detection using graph and spatio-temporal models accelerated on commodity GPUs, (2) deterministic, auditable deployment and lifecycle management of monitoring pipelines through IaC, (3) model-based self-correcting controls and fault-tolerant supervisory strategies aligned to ASHRAE standards, and (4) organizational preparedness and recovery planning to close the loop between technical detection, operational response, and disaster resilience. Methodological exposition details the system architecture, data handling, algorithmic choices, and software lifecycle practices; a descriptive results section synthesizes expected outcomes and system behavior under a range of fault modes; and an extended discussion elaborates theoretical implications, limitations, and a pathway for future research and deployment. The design emphasizes safety, reproducibility, and operational viability: detection precision and recall are framed alongside latencies achievable by GPU acceleration and the governance benefits afforded by IaC. The conclusions highlight how coordinated technical and organizational practices can materially improve the readiness, response, and recovery of building systems facing sensor faults, actuator failures, or anomalous dynamics during emergencies. This research contributes a pragmatic, integrative roadmap for researchers, facility engineers, and system integrators seeking to move beyond isolated algorithms toward production-ready resilience for critical built-environment infrastructure.

Keywords: GPU-accelerated anomaly detection, HVAC fault-tolerance, infrastructure-as-code, self-correcting controls, building resilience, spatio-temporal monitoring, disaster preparedness

INTRODUCTION

Buildings are socio-technical systems that combine physical infrastructure, control logic, sensor networks, and human operators. The operational integrity of heating, ventilation, and air-conditioning (HVAC) subsystems has profound implications for occupant health, energy use, and the resilience of critical facilities during emergencies (ASHRAE, 2018; ASHRAE, 2020). Contemporary buildings increasingly rely on dense sensor deployments, networked controllers, and software platforms that collect and act upon large volumes of telemetry. While these evolutions enable fine-grained control and energy optimization, they also create new failure modes and dependencies: sensor drifts, communication concurrency anomalies in heterogeneous compute stacks, misconfigurations introduced during software updates, and errors propagated through automated deployment pipelines (Bengea et al., 2015; Alglave et al., 2015; Chinamanagonda, 2019). The complex interplay of cyber-physical components raises pressing questions: how can designers scale detection systems to monitor thousands of signals in real time while ensuring detection robustness, and how can organizations operationalize automated responses that preserve safety and continuity in emergencies

(Alexander, 2015)?

Recognizing these gaps, the present work synthesizes four converging domains: (1) high-throughput machine learning and anomaly detection techniques—especially spatio-temporal graph models—that have demonstrated efficacy in monitoring large distributed sensor arrays (Asres et al., 2023); (2) GPU technology and diagnostic automation which enable factory-grade throughput for large-scale telemetry processing and can reduce detection latency for online monitoring (Alglave et al., 2015; Lulla et al., 2025); (3) infrastructure-as-code (IaC) practices which make deployment auditable, repeatable, and less prone to configuration drift (Chinamanagonda, 2019; Bhattacharjee, 2020); and (4) domain-specific self-correcting control strategies and standards for HVAC systems that provide an operational pathway for translating detection outputs into safe corrective actions (Bengea et al., 2015; Brambley et al., 2011; ASHRAE, 2018). Each domain contributes essential capabilities: spatio-temporal analytics locate and contextualize anomalies across zones and subsystems, GPUs supply the computational headroom for real-time inference, IaC governs the software lifecycle to reduce deployment risks, and self-correcting controls implement safe, constrained responses that reduce energy waste and protect occupants.

The literature already furnishes building blocks for this integrative vision. Spatio-temporal and graph-based anomaly detection has shown promise for data-quality monitoring in complex scientific detectors and industrial systems (Asres et al., 2023). GPU concurrency research underscores opportunities and pitfalls when harnessing parallel hardware for low-latency computation, especially in concurrent multi-tenant environments (Alglave et al., 2015). IaC and automated deployment frameworks reduce human error and enable controlled rollouts of analytics pipelines (Chinamanagonda, 2019; Bhattacharjee, 2020). Research on self-correcting HVAC controls demonstrates how model-based approaches can detect sensor faults and command corrective actions to maintain acceptable environmental conditions (Bengea et al., 2015; Brambley et al., 2011). Meanwhile, disaster preparedness literature emphasizes holistic planning and recovery procedures that cover both technical systems and human operations (Alexander, 2015). Despite this rich body of work, the literature lacks a unified, production-ready architecture that links high-throughput anomaly detection, robust deployment practices, and operationally safe self-correction within a disaster-preparedness framework. This gap is consequential: fragmented solutions leave facilities dependent on ad hoc scripts, costly human interventions, or brittle detection algorithms that do not scale.

This article addresses that gap by proposing a unified framework and an accompanying methodological exposition that operationalizes GPU-accelerated spatio-temporal anomaly detection within an IaC-managed deployment that feeds into self-correcting HVAC controllers and organizational recovery pathways. The framework is designed to be technology-agnostic where possible, yet prescriptive about integration patterns and governance controls that materially reduce risk. By tightly coupling detection and correction while preserving human-in-the-loop oversight, the approach aims to enhance both automated responsiveness and auditability—two properties that are essential for safety-critical systems in buildings. The remainder of the article elaborates the theoretical rationale, system architecture, methodological components, expected outcomes, limitations, and future research directions, providing a comprehensive blueprint for researchers and practitioners to implement resilient monitoring and control for built-environment systems.

METHODOLOGY

The methodological exposition describes a systems-oriented design covering architecture, data ingestion and labeling strategies, anomaly detection algorithms, GPU-enabled execution, deployment practices with IaC, and mechanisms for safe corrective actions within HVAC control. Each subcomponent is described in depth to facilitate reproducibility and to articulate the trade-offs encountered when moving from research prototypes to production systems.

System architecture and design principles

The proposed system architecture is guided by four principles: modularity, determinism, safety, and observability. Modularity partitions the architecture into clear subsystems—sensor layer, ingestion and preprocessing, analytics engine, decision manager, actuator interface, and organizational workflows—so that each can be tested, upgraded, and audited independently (Bhattacharjee, 2020). Determinism emphasizes that analytics and control behaviors should be reproducible given the same inputs; IaC, versioned model artifacts, and deterministic GPU kernels are crucial here (Chinamanagonda, 2019). Safety requires that any automated corrective action operate within pre-defined, conservative bounds to avoid exacerbating faults or violating occupant comfort and safety constraints; ASHRAE guidelines provide domain-specific bounds and sequences of safe operation (ASHRAE, 2018; ASHRAE, 2020). Observability mandates rich telemetry at every layer: not only sensor readings but also metadata about model decisions, deployment versions, and control commands are captured for diagnosis and after-action review (Bhattacharjee, 2020).

Sensor layer and data model

Buildings typically host heterogeneous sensors: temperature, humidity, differential pressures, CO_2, damper and valve positions, supply and return air flows, and energy meters. Each sensor reading should be captured with contextual metadata: sensor ID, location (zone), equipment association (AHU, VAV), sampling timestamp, and a quality tag indicating known maintenance history or calibration status. The data model must explicitly encode relationships—spatial adjacency (rooms in same floor), equipment connectivity (VAV boxes served by an AHU), and temporal association (periodic sampling). These relational semantics are the substrate for spatio-temporal graph models (Asres et al., 2023). For instance, a graph node may represent a VAV box or sensor, while edges encode physical or logical connections that inform expected correlation patterns; these graphs are dynamic and must tolerate reconfiguration when zones are repurposed or equipment is replaced (Dong, 2019; Asres et al., 2023).

Ingestion, normalization, and quality checks

Robust ingestion pipelines must gracefully handle delays, out-of-order messages, and partial outages. The design uses tiered buffering with watermarking semantics so near-real-time analytics operate on consistent views while allowing late-arriving data to be reconciled (Bhattacharjee, 2020). Normalization maps vendor-specific signal encodings to canonical units and scales, and applies device-specific calibrations where available. Early-stage quality checks implement lightweight heuristics—range checks, sudden jump detection, and plausibility gates—to flag obviously corrupted inputs. These gates serve both to protect downstream models from garbage data and to produce candidate fault flags that may be processed differently from algorithmic anomalies (Bengea et al., 2015).

Graph-based spatio-temporal anomaly detection

At the analytic core sits a spatio-temporal anomaly detector constructed from graph neural networks (GNNs) and temporal encoders that explicitly model inter-sensor relationships and dynamic behaviors. Graph representations capture spatial structure and cross-sensor dependency statistics; temporal modules—such as gated recurrent units or temporal convolutions—model expected dynamics for each node and for aggregated equipment behaviors. The anomaly detector operates in two complementary modes: (1) unsupervised, density- or reconstruction-based detection that flags deviations from learned normal manifolds; and (2) supervised, classifier-based detection that incorporates labeled fault signatures where available (Asres et al., 2023; Dong, 2019).

Unsupervised detection: The unsupervised model learns a latent representation of normal operational patterns in which reconstruction error or predictive surprisal indicates anomalies. Training uses historical telemetry segmented by operational regimes (occupied/unoccupied, seasonal setpoints) to prevent conflation across modes. Normalizing flows or variational autoencoder variants permit probabilistic scoring and uncertainty quantification for each node and timestamp (Asres et al., 2023).

Supervised detection and hybridization: Where curated fault datasets exist (e.g., sensor bias, damper stuck, valve leak), supervised classifiers enhance detection specificity. Hybrid architectures combine unsupervised scoring as a prescreen with supervised models for confirmation. Hybridization reduces false positives while preserving sensitivity to novel anomalies (Dong, 2019).

Explainability and root-cause prioritization: Practical deployments require not only a binary anomaly label but also interpretable explanations linking anomalies to likely root causes (sensor drift, network latency, systemic control oscillation). Feature attribution techniques and graph-based attention mechanisms can highlight which nodes and features most contributed to the anomaly score, enabling prioritized inspection and targeted corrective actions (Asres et al., 2023).

GPU acceleration and concurrency considerations

A central tenet of the framework is that real-time monitoring at building-portfolio scale requires significant compute. Commodity GPUs offer high-throughput capability for both training and low-latency inference of deep spatio-temporal models. However, deploying GPU-accelerated pipelines safely requires attention to concurrency semantics and runtime determinism (Alglave et al., 2015). Concurrency anomalies can arise from non-deterministic scheduling, mixed-precision arithmetic, or preemption in shared GPU environments; thus system architects must design kernels and runtimes with reproducibility tests, fixed random seeds for inference where appropriate, and isolation policies for multi-tenant workloads (Alglave et al., 2015; Lulla et al., 2025).

Batching and micro-batching: The inference layer uses adaptive micro-batching to trade throughput and latency. During high event rates (e.g., an AHU startup), the system reduces batching delay to maintain low detection latency; during stable periods, larger batches improve GPU utilization without affecting responsiveness (Lulla et al., 2025).

Edge vs. cloud trade-offs: For facilities with stringent latency or network constraints, edge GPUs co-located with building controllers provide localized inference and reduce dependency on cloud connectivity. Cloud-based GPUs provide elastic capacity for cross-building correlation analysis and model retraining. IaC governs both edge and cloud provisioning to ensure consistency across environments (Chinamanagonda, 2019).

Model lifecycle, versioning, and governance

Model artifacts are treated as first-class configuration items. Each model version is packaged with metadata: training data snapshot, hyperparameters, evaluation metrics, expected operational regimes, and human-reviewed decision thresholds. Model registries enable audited rollbacks, A/B testing, and staged rollouts. The IaC pipeline automates deployment of model packages to target compute (edge devices or cloud instances), applies configuration templates, and orchestrates canary releases to mitigate risks from model regressions (Bhattacharjee, 2020).

Infrastructure-as-code and automated deployment

IaC codifies the entire monitoring and analytics stack—compute instances, container orchestration, network policies, storage configuration, and logging—into declarative manifests (Chinamanagonda, 2019). Version-

controlled IaC templates provide reproducible builds, enable peer review of infrastructure changes, and reduce ad hoc manual steps that historically introduce configuration drift. IaC templates include test harnesses for simulating sensor patterns, synthetic fault injections, and integration tests that validate the end-to-end detection-to-action pipeline before promotion to production (Bhattacharjee, 2020).

Testing and continuous integration: Continuous integration pipelines run unit and integration tests for data ingestion, model inference, and actuator command flows. Synthetic fault-injection tests simulate common HVAC faults as well as compound failures (e.g., sensor bias plus communication drop) to verify detection accuracy and safe action sequencing (Brambly et al., 2011; Dexter & Pakanen, 2001).

Decision manager and safety policies

The decision manager mediates between detection outputs and actuators. It enforces safety policies, operator preferences, and escalation procedures. Policies include conservative bounds on actuator commands, rollback triggers if corrective actions worsen conditions, and human-in-the-loop thresholds where automatic intervention is allowed only after operator acknowledgement (Bengea et al., 2015). Policies are encoded in declarative rule sets that can be formally verified for simple invariants (e.g., maximum valve opening percentage during occupancy) and are version-controlled with IaC.

Self-correcting control strategies

Self-correcting control strategies implement model-based control actions designed to maintain service while minimizing risk. These include sensor fusion to mitigate single-sensor failures, control reconfiguration to operate in a degraded-but-safe mode, and optimization of sequences for recovery that avoid large transients (Fernandez et al., 2009a; Bengea et al., 2015).

Sensor fusion and virtual sensors: Virtual sensors combine redundant measurements and physical models to estimate quantities when a direct sensor fails. For example, supply air temperature can be estimated from return temperature and measured flow rates, conditioned by known coil behavior. Virtual sensors reduce unnecessary actuations while providing continuity for control loops (Fernandez et al., 2009a).

Degraded mode operation: When the decision manager confirms an actuator or sensor fault, control logic shifts to a pre-defined degraded mode that maintains occupant safety: limiting maximum supply flow, setting conservative temperature setpoints, and increasing ventilation to mitigate indoor air quality concerns if dampers are stuck. Degraded modes are documented and aligned with ASHRAE guidance (ASHRAE, 2018).

Human-in-the-loop escalation: Not all anomalies are safe to resolve automatically. The system delineates classes of anomalies—those safe for automated correction, those requiring operator approval, and those that demand immediate manual intervention. Escalation channels, pre-scripted action sets, and operator dashboards support coordinated responses and maintain compliance with emergency plans (Alexander, 2015).

Organizational preparedness and post-fault recovery

Technical systems are nested within organizational procedures for emergency preparedness and recovery. Detection and self-correction capabilities must be embedded in disaster planning cycles, drills, and after-action review processes. Recovery playbooks specify roles, responsibilities, and decision thresholds; observational data captured by the monitoring system informs root-cause investigations and continuous improvement (Alexander, 2015). Regular training exercises that include simulated sensor failures and automation misbehaviors help maintain team readiness and reveal gaps between automation capabilities and operational practice.

Evaluation metrics and monitoring the monitor

Operational success depends on well-chosen evaluation metrics: detection precision/recall, mean time to detection, mean time to recovery, false alarm rate per sensor per day, control stability after automated action, and business-impact metrics such as maintained occupant comfort hours and energy deviation from baseline. Additionally, the system monitors its own health—data ingestion latencies, model inference times, GPU utilization, and IaC drift—so that failures in the monitoring pipeline are themselves detectible and can be triaged (Bhattacharjee, 2020).

Ethical, privacy, and governance considerations

Telemetry from buildings can carry privacy-sensitive information (e.g., occupancy patterns). Data governance policies must define retention, anonymization, and access controls. Automated control actions must respect explicit safety constraints and be auditable. Institutional review and stakeholder engagement help ensure responsible deployment and alignment with building-user expectations (Alexander, 2015).

RESULTS

This section presents a descriptive synthesis of expected outcomes when the proposed framework is implemented in practice. Given the constraints of this article (no primary experimental dataset), results are articulated as projected performance characteristics, qualitative behaviors under fault scenarios, and quantified estimates grounded in contemporary literature on GPU-enabled analytics, self-correcting HVAC controls, and automated deployment practices.

Throughput and latency improvements from GPU acceleration

GPU-accelerated inference for spatio-temporal models substantially reduces per-batch inference latency and enables higher input throughput compared to CPU-only deployments. Prior work indicates that GPU acceleration can enable real-time inference for deep models at millisecond-to-subsecond latencies depending on model complexity and batch strategy (Alglave et al., 2015; Lulla et al., 2025). Practically, a deployment using commodity data-center GPUs with optimized kernels can handle hundreds of sensor nodes with sub-second latency for each inference pass when micro-batching strategies are used. This latency is sufficient for online detection of abrupt events (e.g., damper stuck at startup) and near-real-time scoring of slowly developing anomalies (e.g., sensor drift).

Detection accuracy and hybrid modeling

Hybrid detection architectures combining unsupervised reconstruction models and supervised classifiers are expected to yield high sensitivity to novel anomalies while preserving specificity for known fault classes. Unsupervised models capture emergent deviations that do not match historical patterns, and supervised modules reduce false positives for frequently observed faults by leveraging labeled examples (Dong, 2019; Asres et al., 2023). In practice, one can expect unsupervised detectors to provide early warnings, while supervised components reduce confirmation time for specific corrective workflows.

Operational behavior under common fault modes

The integrated system demonstrates characteristic behaviors across fault scenarios:

Sensor bias: When a temperature sensor drifts, unsupervised detectors flag the divergence in reconstruction error relative to neighboring sensors and virtual-sensor estimates. The decision manager cross-checks with

virtual sensor estimates and, if confidence is high, triggers a recalibration workflow or switches the control loop to use the virtual sensor until physical calibration is completed (Bengea et al., 2015; Fernandez et al., 2009a).

Damper stuck: A stuck damper is typically detectable by comparing commanded position to feedback position, correlated with flow and pressure measurements. Early detection via graph models identifies correlated anomalies across downstream VAV boxes and initiates a degraded ventilation strategy that maintains minimum ventilation rates while limiting energy impacts (Bengea et al., 2015; Brambley et al., 2011).

Communication loss: Intermittent telemetry loss manifests as missing nodes or elevated latency. The ingestion layer's watermarking and buffering provide resilience to transient losses; the decision manager marks affected control loops as in degraded mode and escalates to operators if persistent. IaC-enabled deployment of redundant network paths (when available) facilitates rapid reconfiguration (Chinamanagonda, 2019).

Control oscillations: Oscillatory dynamics are often the result of mis-tuned controllers or delayed feedback. Graph-temporal models can detect abnormal spectral signatures and phase relationships between actuation and sensor response. The system can temporarily limit controller gain via a safe policy to damp oscillations while alerting control engineers for retuning (Bengea et al., 2015).

Human-machine workflows and recovery

By codifying policies for automatic vs. manual interventions, the system preserves operator oversight in complex or safety-critical situations (Alexander, 2015). Automated corrective actions are conservative and reversible; operators receive contextualized explanations and are able to accept or override actions. After recovery, the system's audit logs and diagnostic outputs support root-cause analysis and continuous improvement. IaC ensures the deployed remediation scripts and rollback mechanisms are traceable to a versioned manifest, easing compliance and post-incident review (Bhattacharjee, 2020).

Resilience and readiness for disasters

Embedding the monitoring architecture within disaster preparedness plans materially enhances organizational resilience. Early detection of faults that would degrade HVAC performance during extreme weather or emergency occupancy surges reduces the likelihood that systems exacerbate harm. The combination of automated detection, conservative autonomous control, and well-practiced human escalation pathways leads to faster stabilization and clearer after-action evidence that informs remediation and policy updates (Alexander, 2015).

Governance and lifecycle oversight

The use of IaC and model registries improves governance and reduces the risk of silent regressions. Rollback capabilities, staged rollouts, and canary testing reduce the probability that a model update or infrastructure change produces destructive behavior. Continuous monitoring of model performance detects concept drift and triggers retraining, which, when coupled with versioned artifacts, preserves accountability (Bhattacharjee, 2020).

Energy and comfort trade-offs

Automated fault mitigation strategies prioritize occupant safety and comfort, which can sometimes increase short-term energy use (e.g., increasing ventilation to counteract a stuck damper). However, early detection

and targeted corrective actions often reduce prolonged inefficiencies that otherwise accumulate when faults go undetected. Studies of self-correcting controls show that proactive fault detection and correction can yield net energy savings over time, even when temporary conservative actions introduce short-term energy costs (Bengea et al., 2015; Economidou, 2011).

DISCUSSION

This discussion interprets the methodological choices and projected results, examines theoretical implications, considers limitations and potential failure modes, and outlines a measured roadmap for future research and real-world deployment.

Interpreting the integrated approach

The proposed integration blurs traditional boundaries between analytics, infrastructure management, and control engineering. Historically, building analytics and controls were separate: analytics lived in research labs or vendor cloud services, while controllers were embedded in building management systems with hard-coded logic. The architecture described here treats analytics as an operational subsystem with deterministic deployment practices and governance, aligning analytics behavior closely with control actions through the decision manager and safety policies. This co-design—analytics and control specified together—enables more confident automation while preserving auditable human oversight (Bhattacharjee, 2020; Bengea et al., 2015).

The role of GPUs and concurrency concerns

GPUs are more than a performance acceleration; they change the unit economics of real-time monitoring. With sufficient compute, teams can deploy more sophisticated models, perform cross-building correlation, and maintain ensembles that adapt to seasonal regimes. However, GPU environments introduce non-trivial risks: non-deterministic execution and concurrency bugs (Alglave et al., 2015). The system must therefore treat GPU runtimes with the same engineering rigor as other safety-critical software: deterministic kernels when necessary, strict isolation for production loads, and reproducible configuration templates via IaC (Lulla et al., 2025). Multi-tenant deployment scenarios—common in managed service models—must be architected to prevent interference that could delay detection in critical facilities.

Model reliance and human oversight

While advanced models can detect subtle anomalies, overreliance on automation without clear human-in-the-loop boundaries risks brittleness in unforeseen states. The decision manager's policy classification—automatic, assisted, manual—represents an essential compromise: automation accelerates routine recoveries, while operators remain central for ambiguous or high-stakes situations (Alexander, 2015). Furthermore, explainability mechanisms are critical: operators must understand why a model recommended a given action, especially when the action affects safety or regulatory compliance.

Limitations and potential failure modes

Data scarcity and label paucity: Supervised models require labeled fault examples which are often scarce; synthetic fault injection and transfer learning can help but may not capture all real-world behavior (Dong, 2019). Unsupervised approaches mitigate label scarcity but may produce higher false positive rates without careful tuning and stratified training across operational regimes.

Sensor and equipment heterogeneity: Building fleets commonly exhibit heterogeneity in sensors, protocols (e.g., BACnet), and equipment vintage (ASHRAE, 2020). Heterogeneity complicates canonical normalization

and demands flexible ingestion layers and per-site calibration. IaC simplifies reproducible deployments but cannot wholly eliminate hardware variability.

Operational constraints: Not all facilities have on-premise GPUs or reliable network connectivity. The framework accommodates edge deployment but requires capital investment and operational competencies that may be beyond some organizations. Cloud services can offset this but may introduce latency and privacy trade-offs.

Adversarial and security concerns: Sensor spoofing or adversarial inputs could mislead detection systems. Strong authentication, cryptographic integrity checks, and cross-validation across independent modalities (e.g., energy meter anomalies corroborating sensor readings) reduce this risk. Security practices must be integrated into IaC manifests and runtime configurations (Chinamanagonda, 2019).

Governance friction and regulatory factors: Automated control actions that influence occupant safety or environmental conditions may be subject to regulations or institutional policies. Legal and compliance teams must be part of deployment planning; explicit documentation and operator oversight mitigate liability (Alexander, 2015).

Pathways for future research

Benchmark datasets and shared tasks: The field would benefit from standardized, richly annotated datasets that capture a broad set of fault modes across building types and climates. Shared benchmarks enable comparative evaluation of detection algorithms and facilitate progress in supervised learning for rare fault classes (Dong, 2019).

Hybrid human-AI collaboration studies: Empirical research should evaluate how operators interact with automated recommendations in real-world settings. Observational studies and controlled trials can reveal effective escalation thresholds, interface designs for explanation, and training protocols that enhance trust and effectiveness (Alexander, 2015).

Formal verification for safety policies: Research into formal methods for verifying critical safety invariants encoded in decision manager policies could reduce the risk of unsafe automated actions. Lightweight formal proofs and property checking could be integrated into IaC pipelines to ensure safety constraints are preserved during updates.

Adversarial robustness and secure telemetry: Research is needed to harden anomaly detectors against spoofing and adversarial perturbations. Multi-modal corroboration, cryptographic attestation of sensor firmware, and secure telemetry channels are promising directions.

Operational economics and lifecycle studies: Longitudinal studies that quantify total cost of ownership, energy impacts, and operational savings from early detection can help organizations justify investments in GPU-enabled monitoring and IaC practices (Economidou, 2011).

Recommendations for practitioners

Start small and iterate: Pilot deployments focused on critical zones (e.g., data centers, hospital wards) yield early value and allow teams to refine models and policies before broader rollouts.

Invest in data hygiene and metadata: High-quality metadata and consistent sensor naming conventions considerably simplify modeling and enable transferability across sites.

Integrate IaC from day one: Treat infrastructure, model artifacts, and operator playbooks as version-controlled artifacts to enable reproducibility and auditability.

Prioritize explainability and operator training: Automated systems should provide concise, actionable explanations and be accompanied by training that clarifies when automation is expected to act and when manual intervention is required.

Adopt conservative default policies: Default automatisms should err on the side of safety—limited commands and reversible actions—until teams build confidence through measured success.

CONCLUSION

Modern building systems require a coordinated synthesis of advanced analytics, robust infrastructure practices, and operationally safe control strategies. This article presented a comprehensive framework that integrates GPU-accelerated spatio-temporal anomaly detection, infrastructure-as-code deployment, self-correcting HVAC controls, and disaster-preparedness planning. By articulating a full-stack design—from sensor metadata through decision manager policies to organizational recovery playbooks—the framework reconciles the needs for real-time responsiveness, reproducibility, safety, and governability. GPU acceleration unlocks real-time analytics at scale while IaC ensures deterministic, auditable deployments; self-correcting controls translate detection into safe remedial actions; and preparedness planning embeds these technical capabilities in organizational workflows that support resilience during emergencies. While challenges remain—data scarcity, heterogeneity, security risks, and governance complexity—this synthesis provides a pragmatic roadmap for moving from isolated laboratory prototypes to production-ready resilience for critical built-environment systems. The research agenda ahead includes creating shared fault datasets, empirical human-AI collaboration studies, formal verification of safety policies, and economic analyses that clarify the return on investment. Realizing the vision will require interdisciplinary collaboration among control engineers, data scientists, facilities operators, and policy makers. When these actors coordinate, the resulting systems can materially improve the safety, comfort, and energy efficiency of buildings while strengthening community resilience against disruptions.

REFERENCES

1. Alexander, D. E. (2015). Disaster and emergency planning for preparedness, response, and recovery. Oxford University Press.

2. Alglave, J., Batty, M., Donaldson, A. F., Gopalakrishnan, G., Ketema, J., Poetzl, D., ... & Wickerson, J. (2015). GPU concurrency: Weak behaviours and programming assumptions. ACM SIGARCH Computer Architecture News, 43(1), 577-591.

3. Asres, M. W., Omlin, C. W., Wang, L., Yu, D., Parygin, P., Dittmann, J., ... & Cms-Hcal Collaboration. (2023). Spatio-temporal anomaly detection with graph networks for data quality monitoring of the Hadron Calorimeter. Sensors, 23(24), 9679.

4. Bengea, S. C., Li, P., Sarkar, S., Vichik, S., Adetola, V., Kang, K., Lovett, T., Leonardi, F., Kelman, A.D. (2015). Fault-tolerant optimal control of a building HVAC system. Science and Technology for the Built Environment, 21(6), 734–751.

5. Bhattacharjee, A. (2020). Algorithms and Techniques for Automated Deployment and Efficient Management of Large-Scale Distributed Data Analytics Services (Doctoral dissertation, Vanderbilt University).

6. Brambley, M., Fernandez, N., Wang, W., Cort, K.A., Cho, H., Ngo, H., Goddard, J.K. (2011). Final project report: Self-correcting controls for VAV system faults filter/fan/coil and VAV box sections. No. PNNL-20452; Pacific Northwest, National Laboratory (PNNL), Richland, WA, USA.

7. Chavan, A. (2022). Importance of identifying and establishing context boundaries while migrating from monolith to microservices. Journal of Engineering and Applied Sciences Technology, 4, E168. http://doi.org/10.47363/JEAST/2022(4)E168

8. Chinamanagonda, S. (2019). Automating Infrastructure with Infrastructure as Code (IaC). Available at SSRN 4986767.

9. Deep, A. T. (2024). Advanced financial market forecasting: integrating Monte Carlo simulations with ensemble Machine Learning models.

10. Dexter, A., Pakanen, J. (Eds.). (2001). Demonstrating Automated Fault Detection and Diagnosis Methods in Real Buildings. Technical Research Centre of Finland, Finland.

11. Dong, M. (2019). Combining unsupervised and supervised learning for asset class failure prediction in power systems. IEEE Transactions on Power Systems, 34(6), 5033-5043.

12. Economidou, M. (2011). Europe's buildings under the microscope. A country-by-country review of the energy performance of buildings. Technical Report Buildings Performance Institute Europe.

13. Fernandez, N.; Brambley, M.; Katipamula, S. (2009a). Self-correcting HVAC controls: Algorithms for sensors and dampers in air-handling units, PNNL-19104; Pacific Northwest; National Laboratory: Richland, WA, USA.

14. Goel, G., & Bhramhabhatt, R. (2024). Dual sourcing strategies. International Journal of Science and Research Archive, 13(2), 2155. https://doi.org/10.30574/ijsra.2024.13.2.2155

15. Lulla, K., Chandra, R., & Ranjan, K. (2025). Factory-grade diagnostic automation for GeForce and data centre GPUs. International Journal of Engineering, Science and Information Technology, 5(3), 537-544.

16. Dhanagari, M. R. (2024). Scaling with MongoDB: Solutions for handling big data in real-time. Journal of Computer Science and Technology Studies, 6(5), 246-264. https://doi.org/10.32996/jcsts.2024.6.5.20

17. ASHRAE. (2018). Guideline 36–2018. High Performance Sequences of Operation for HVAC Systems. ASHRAE, Akron, OH, USA.

18. ASHRAE. (2020). Standard 135-2020—BACnetTM—A Data Communication Protocol for Building Automation and Control Networks. Available online: https://www.ashrae.org/technical-resources/bookstore/bacnet (accessed on 10 Aug 2021).

19. Bengea, S.C., Li, P., Sarkar, S., Vichik, S., Adetola, V., Kang, K., Lovett, T., Leonardi, F., Kelman, A.D. (2015). Fault-tolerant optimal control of a building HVAC system. Science and Technology for the Built Environment, 21(6), 734–751.

20. Brambley, M., Fernandez, N., Wang, W., Cort, K.A., Cho, H., Ngo, H., Goddard, J.K. (2011). Final project report: Self-correcting controls for VAV system faults filter/fan/coil and VAV box sections. No. PNNL-20452; Pacific Northwest, National Laboratory (PNNL), Richland, WA, USA.

21. Dexter, A., Pakanen, J. (Eds.). (2001). Demonstrating Automated Fault Detection and Diagnosis Methods in Real Buildings. Technical Research Centre of Finland, Finland.

22. Fernandez, N.; Brambley, M.; Katipamula, S. (2009a). Self-correcting HVAC controls: Algorithms for sensors and dampers in air-handling units, PNNL-19104; Pacific Northwest; National Laboratory: Richland, WA, USA.