# Leveraging Data Science for Real-Time Threat Intelligence and Anomaly Detection in Critical Infrastructure

**Author:** Haruto Sato **Affiliation:** Department of Robotics, University of Tokyo (Japan)

**Email:** haruto.sato@u-tokyo.ac.jp

**Abstract**

Critical infrastructure systems (power, water, transportation, healthcare, and communications) are increasingly digitized and interconnected, which improves efficiency but also expands attack surfaces. Real-time threat intelligence and anomaly detection using advanced data science techniques are essential to detect, interpret, and respond to malicious activity and failures before they cascade into large-scale outages or safety incidents. This paper presents a comprehensive, research-grade synthesis of theoretical foundations, architectures, algorithms, evaluation methodologies, and deployment considerations for real-time threat intelligence and anomaly detection applied to critical infrastructure. We integrate classical statistical approaches, machine learning (ML), deep learning (DL), streaming analytics, graph analytics, and explainable AI (XAI) to construct a practical yet rigorous blueprint for systems that require high availability, low latency, interpretability, and regulatory compliance. Case studies and example designs for power grids, industrial control systems (ICS)/SCADA, transportation networks, and healthcare cyber-physical systems (CPS) illustrate tradeoffs between detection accuracy, false alarm rates, computational cost, and explainability. We conclude with an agenda for future research, including causal anomaly detection, federated/edge analytics for privacy and latency, and formal verification of ML components.

**Keywords:** critical infrastructure, anomaly detection, real-time analytics, threat intelligence, explainable AI, streaming machine learning, cyber-physical systems

## 1. Introduction

### 1.1 Motivation

Critical infrastructure forms the backbone of modern society. As these systems become more data-driven and interconnected, the risks associated with cyber attacks, insider threats, cascading failures, and sensor faults grow. Incidents such as large-scale power outages caused by cyber intrusion, or water treatment manipulations, demonstrate that adversaries can exploit digital controls to cause physical harm. Traditional signature-based intrusion detection and periodic manual audits are insufficient for timely identification of novel attacks and subtle anomalies. Real-time threat intelligence—combining high-velocity data collection, automated anomaly detection, and rapid prioritization—is therefore vital to protect safety, availability, and resilience.

### 1.2 Scope and contributions

This article synthesizes contemporary academic and applied knowledge into a thorough, submission-ready manuscript that:

1. Frames requirements unique to critical infrastructure (low latency, interpretability, high reliability).
2. Surveys and unifies algorithmic approaches for real-time anomaly detection and threat intelligence.
3. Proposes architectures and streaming pipelines optimized for CPS/ICS contexts.
4. Discusses evaluation metrics, datasets, and experimental design to yield reproducible research.
5. Integrates explainability and human-in-the-loop designs to enable operator trust and regulatory compliance.
6. Presents concrete case studies with recommended model selections and deployment pathways.

We place special emphasis on the balance between detection performance and operational constraints and highlight open problems for future research.

## 2. Background and Related Work

### 2.1 Definitions and taxonomy

We adopt the following working definitions: an *anomaly* is an observation or pattern that deviates from expected behavior given a context; a *threat* is a potential or attempted action with malicious intent; *real-time* implies actionable detection within an operational tempo required by the system (often milliseconds to seconds in control loops, and seconds to minutes for supervisory monitoring). Threat intelligence is the pipeline from data ingestion → detection → context enrichment → prioritization → response orchestration.

Taxonomically, anomaly detection approaches can be grouped as:

- **Statistical** (parametric and non-parametric),
- **Distance/neighborhood-based** (k-NN, LOF),
- **Density-based** (e.g., Gaussian mixture models),
- **Model-based** (HMMs, ARIMA),
- **Machine learning** (supervised, semi-supervised, unsupervised),
- **Deep learning** (autoencoders, LSTMs, CNNs for temporal/spatial patterns),
- **Graph/network analytics** (community detection, graph neural nets),
- **Hybrid systems** (ensemble and multi-stage).

Comprehensive surveys such as Chandola et al. (2009) provide foundational overviews of statistical and ML approaches for anomaly detection across domains (Chandola, Banerjee, & Kumar, 2009). The particular challenges of ML for network intrusion detection and adversarial adaptation are discussed in Sommer & Paxson (2010) and follow-on work.

### 2.2 Anomaly detection in networked control systems and CPS

Networked control systems (NCS) and CPS introduce correlated multi-modal signals (sensor telemetry, actuator commands, network flows, logs). Lakhina et al. (2004) demonstrated that network traffic anomalies often exhibit low-dimensional structures exploitable by subspace methods—techniques that translate well to ICS telemetry. The literature identifies three broad challenge families: (1) data heterogeneity and scale, (2) real-time inference under resource constraints, and (3) explainability and human trust.

### 2.3 Real-time analytics and streaming ML

Streaming data frameworks (e.g., Apache Flink, Kafka Streams) and online learning algorithms permit continuous model updates and low-latency inference necessary for operational deployments. These frameworks, combined with model types such as online gradient descent, incremental clustering, and reservoir computing, produce systems that adapt to concept drift and evolving adversary tactics.

### 2.4 Explainability and human-centered requirements

Explainable AI (XAI) is critical in safety-sensitive domains. Ozdemir & Fatunmbi (2024) articulate methods for bridging predictive accuracy with interpretability, particularly in healthcare; the same principles apply to critical infrastructure where operators require concise rationales for alerts to avoid alarm fatigue and enable rapid remediation. We incorporate XAI strategies—feature attribution, rule extraction, and example-based explanations—into recommended pipelines (Ozdemir & Fatunmbi, 2024).

## 3. Requirements for Critical Infrastructure Detection

### 3.1 Operational constraints

Critical infrastructure detection systems must satisfy:

- **Low latency:** Detection and enrichment within operational time windows.
- **High reliability and availability:** Fault tolerance and graceful degradation.
- **Low false positive rate (FPR):** To prevent operator overload and unnecessary intervention.
- **Robustness to concept drift and adversarial manipulation.**
- **Explainability and auditable decision trails** for operators and regulators.

- **Privacy and data governance** compliance, particularly in healthcare and transportation.

### 3.2 Data characteristics

Data streams include telemetry (time series), discrete events (logs), network flows, topological state (graph), and third-party threat intelligence feeds. Data quality issues—missingness, timestamp skew, sensor drift—must be addressed prior to modeling.

### 3.3 Threat models

Systems should consider adversary capabilities: stealthy sensor manipulation, replay attacks, supply-chain tampering, and false data injection. Detection strategies should therefore combine behavioral baselines with invariants derived from physical laws and process models.

## 4. Architecture and System Design

### 4.1 High-level architecture

We propose a modular, layered architecture:

1. **Data ingestion layer:** Collect streams from sensors, network taps, logs, and external TI (threat intelligence) sources. Ensure reliable, ordered transport (e.g., Kafka).
2. **Preprocessing and normalization:** Synchronize timestamps, resample, handle missing values, and perform feature extraction (spectral features, derivatives, statistical aggregates, graph features).
3. **Feature store / context enrichment:** Maintain short-term, low-latency feature stores for operational models and longer history for retraining and post-incident analysis.
4. **Detection engine:** Real-time models (lightweight anomaly detectors, ensemble models) for immediate inference; batch/near-real-time models (deep sequence models) for complex patterns.
5. **Score fusion and prioritization:** Combine detectors with rule-based and threat intelligence signals into a risk score; prioritize alerts.
6. **Explainability & visualization:** Provide concise explanations and recommended remediation actions to operators.
7. **Response orchestration:** Trigger automated containment (e.g., isolate a device) or human workflow for investigation.
8. **Feedback & learning loop:** Human analyst feedback used for labeling and incremental model updates.

Figure 1 (conceptual) [*omitted here*] would show data flow between layers.

### 4.2 Edge vs. cloud tradeoffs

Edge processing reduces latency and bandwidth use but imposes resource constraints; cloud enables heavy models and aggregation across sites but adds communication latency and potential attack surface. We recommend a hybrid design: initial filtering and simple detectors at edge; complex correlation, graph analytics, and model retraining in cloud or regional data centers.

### 4.3 Security and reliability considerations for the detection stack

The detection infrastructure itself must be hardened: encrypted channels, role-based access controls, cryptographic attestation for model updates, and secure logging. Consider Byzantine-robust aggregation and defenses against poisoned training data.

## 5. Data Preparation and Feature Engineering

### 5.1 Time series pre-processing

Key steps: timestamp alignment, outlier smoothing, trend/seasonality decomposition (STL), normalization, and windowing strategies (sliding/hopping windows). For streaming, use online algorithms that compute aggregates incrementally.

### 5.2 Multi-modal feature extraction

- **Statistical features:** mean, variance, skewness, kurtosis, quantiles in sliding windows.
- **Temporal features:** autocorrelation, partial autocorrelation, spectral power, wavelet coefficients.
- **Graph features:** centrality measures, community membership, motifs counts for networked systems.
- **Protocol features (network):** packet sizes, interarrival times, flags, entropy of payload sizes.
- **Control-model features:** residuals between observed sensors and model-predicted outputs (state estimator residuals).

Physical invariants are invaluable: energy conservation, mass balance, or control loop setpoint invariants can generate residuals that highlight manipulations.

### 5.3 Labeling strategies and weak supervision

In many CPS domains labeled anomalous data are scarce. We recommend weak supervision, synthetic injection of attack scenarios (red-team), unsupervised pretraining, and active learning where operators label uncertain events.

## 6. Algorithms for Real-Time Detection

### 6.1 Lightweight statistical and rule-based detectors (Tier 1)

Purpose: filter obvious anomalies with minimal latency.

- Thresholds, EWMA/CUSUM, z-score and change point detection (e.g., Bayesian online changepoint detection).
- Benefits: interpretable, small memory/compute footprint.
- Limitations: brittle to concept drift and multivariate dependencies.

### 6.2 Unsupervised learning (Tier 2)

- **k-means / streaming k-means:** incremental clustering to find outliers.
- **Local Outlier Factor (LOF):** density-based anomaly score.
- **Isolation Forest (streaming variants):** efficient isolation for high-dimensional data.

These can run continuously and flag events for deeper analysis.

### 6.3 Deep sequence models and representation learning (Tier 3)

- **Autoencoders (AE, Variational AE):** unsupervised reconstruction error as anomaly measure; useful for multi-sensor correlation modeling.
- **Recurrent Neural Networks (LSTM), Temporal Convolutional Networks (TCN):** model temporal dependencies, good for detecting subtle deviations in sequences.
- **Transformer architectures for time series:** applying attention to capture long-range dependencies.
- **Graph Neural Networks (GNN):** model relational data between devices, lines, or substations.

Deep models provide richer detection power but need careful calibration for false positive control and explainability.

### 6.4 Hybrid and ensemble methods

Layering detectors (fast, coarse → slow, precise) reduces overall latency while preserving accuracy: e.g., CUSUM triggers LSTM inference; graph heuristics elevate risk score which invokes AE/GNN.

### 6.5 Online and continual learning

Use online gradient updates, mini-batch incremental retraining, reservoir sampling, and meta-learning mechanisms to adapt to evolving baselines without full retraining.

### 6.6 Adversarial robustness

Adversaries may manipulate features or inject poisoned samples. Defenses include robust statistics, anomaly detectors in feature space (to detect poisoning), gradient masking (caveats), and formal verification of critical model components.

## 7. Explainability and Human-in-the-Loop Integration

### 7.1 Why explainability matters

Operators must trust alerts; black-box outputs without provenance are unlikely to be actioned. XAI techniques provide transparency, increase adoption, and facilitate post-incident forensics (Ozdemir & Fatunmbi, 2024).

### 7.2 Techniques

- **Feature attribution:** SHAP, LIME variants adapted for streaming contexts (local explanations).
- **Prototype-based explanations:** retrieving similar historical events that illustrate why an alert was raised.
- **Rule extraction from models:** approximate decision rules (if-then) derived from trained models.
- **Counterfactuals:** minimal changes that would change the model output; helpful for mitigation guidance.
- **Visualization:** timeline heatmaps, graph overlays of implicated devices/flows, and concise human-readable rationale.

### 7.3 Human feedback loop

Design active learning interfaces where analysts can confirm/annotate alerts, which are then used to refine models. Use confidence thresholds to route only high-uncertainty events for human review.

## 8. Evaluation and Metrics

### 8.1 Offline vs. online evaluation

Offline evaluation on labeled datasets is necessary for initial development but insufficient to assess concept drift and real-world constraints. Online A/B testing, shadow deployments, and red-team exercises are essential.

### 8.2 Metrics

- **Detection performance:** precision, recall, F1, area under ROC/PR curves.
- **Operational metrics:** mean time to detection (MTTD), false alarms per day, analyst time per alert.

- **Robustness metrics:** sensitivity to adversarial perturbations, performance under missing data.
- **Resource metrics:** latency, memory footprint, CPU/GPU utilization.

### 8.3 Datasets and benchmarks

Benchmarking suffers due to a lack of public, realistic ICS datasets. We recommend combination of public datasets (where available), sanitized anonymized operational traces, and synthetic but physically plausible scenarios (validated by domain experts). Rigorous reporting should include dataset provenance, labeling process, and availability for reproducibility.

## 9. Case Studies and Applications

### 9.1 Electric power grid: distribution and transmission

**Challenge:** Detect coordinated false data injection attacks on state estimation, and early detection of equipment anomalies.

**Recommended pipeline:** Edge residue monitors for state estimator residuals; AE on multivariate substation telemetry; GNN across grid topology for correlated anomalies; fusion with external threat intelligence (e.g., IP blacklists). Emphasize physically grounded invariants (Kirchhoff laws) to reduce false positives.

### 9.2 Industrial control systems (SCADA)

**Challenge:** High-frequency control loops with safety-critical responses.

**Recommended pipeline:** Lightweight rule engines at PLC level for immediate safety responses; aggregator for sequence modeling (LSTM/TCN) at SCADA historian level; explainable residuals and counterfactuals to advise operator interventions.

### 9.3 Water treatment and distribution

**Challenge:** Spatially distributed sensors with seasonal and operational variability.

**Recommended pipeline:** Hierarchical anomaly detection—local sensor models detect immediate deviations, regional ensemble models detect coordinated anomalies suggestive of contamination or cyber sabotage.

### 9.4 Healthcare/care facilities CPS

**Challenge:** Patient safety and strict privacy.

**Recommended pipeline:** Privacy-preserving analytics (federated learning for device logs), XAI to explain alarms (referencing Ozdemir & Fatunmbi, 2024), and high accuracy models for detecting device tampering or ransomware activity (Fatunmbi, 2024).

*(Specific investigation results, synthetic experiment setups, and hypothetical performance tables would be included in full submission.)*

## 10. Implementation Considerations and Best Practices

### 10.1 Deployment lifecycle

- Start with small, non-intrusive monitoring in shadow mode.
- Validate on historical incidents and red-team tests.
- Adopt gradual rollout with human oversight and rollback mechanisms.
- Maintain model governance: versioning, retraining schedules, and audit trails.

### 10.2 Computational and resource constraints

Use model pruning, quantization, and distillation for edge models. Design tiered models to only invoke heavy inference when needed.

### 10.3 Data governance and privacy

- Anonymize PII before cross-site aggregation.
- Use differential privacy or secure aggregation for federated analytics.
- Maintain auditable access logs for compliance.

### 10.4 Operationalizing explainability

Integrate explanation outputs with operator dashboards and playbooks. Standardize explanation templates (cause, confidence, suggested action, affected assets).

## 11. Limitations, Challenges, and Open Problems

### 11.1 Label scarcity and realistic benchmarks

A major impediment is the scarcity of labeled attack data representative of modern adversaries. Community efforts to share sanitized datasets would accelerate progress.

### 11.2 Adversarial adaptation

Adversaries can adapt to deployed detectors. Continuous evaluation against adaptive attackers and proactive red-teaming remain necessary.

### 11.3 Balancing interpretability with performance

Some high-performing deep models are inherently less interpretable. Research into inherently interpretable models for time series and graphs is required.

### 11.4 Causal detection and intervention planning

Most current systems detect correlation-based anomalies; causal reasoning about root cause and intervention effects is an open, high-impact research area.

## 12. Future Directions

We highlight promising directions:

1. **Causal anomaly detection:** integrating causal models to distinguish root causes from downstream symptoms.
2. **Federated real-time learning:** enabling collaborative learning across organizations without sharing raw data.
3. **Hybrid physics-ML models:** blending first-principles process models with data-driven components for improved robustness.
4. **Formal verification of ML components:** applying formal methods to ascertain safety guarantees.
5. **Adversarial-aware defenses:** ensembles that anticipate evasion tactics and detect model poisoning.
6. **Operational XAI:** concise, standardized explanation formats for operational decision making.

## 13. Conclusion

Protecting critical infrastructure in the real world requires a rigorous, multidisciplinary approach combining domain knowledge (physics, control theory), data science, streaming architectures, and human-centered design. This paper provides a comprehensive blueprint for building real-time threat intelligence and anomaly detection systems tailored to the stringent demands of critical infrastructure. By combining layered detection architectures, explainable methods, and robust evaluation practices, practitioners and researchers can design systems that meaningfully reduce risk while maintaining operator trust and regulatory compliance.

## References

1. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, 19–31.
2. Barford, P., Kline, J., Plonka, D., & Ron, A. (2002). A signal analysis of network traffic anomalies. *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, 71–82.
3. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), Article 15.

4. Dwork, C. (2006). Differential privacy. *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006), Lecture Notes in Computer Science, 4052*, 1–12.

5. Fatunmbi, T. O. (2024). Developing advanced data science and artificial intelligence models to mitigate and prevent financial fraud in real-time systems. World Journal of Advanced Engineering Technology and Sciences, 11(1), 437-456. https://doi.org/10.30574/wjaets.2024.11.1.0024

6. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

7. Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing, 107*, 241–265.

8. Kwon, D., & Lasko, T. (2017). Online change point detection in streaming data. *Journal of Machine Learning Research — Proceedings Track, 68*, 1–8.

9. Lakhina, A., Crovella, M., & Diot, C. (2004). Diagnosing network-wide traffic anomalies. *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, 219–230.

10. Ozdemir, O., & Fatunmbi, T. O. (2024). Explainable AI (XAI) in healthcare: Bridging the gap between accuracy and interpretability. Journal of Science, Technology and Engineering Research, 2(1), 32–44. https://doi.org/10.64206/0z78ev10

11. Rathore, M. M., Ahmad, A., Paul, A., & Rho, S. (2016). Exploiting IoT and big data analytics: Defining cybersecurity solutions for critical infrastructure. *Computer Communications, 98*, 1–14.

12. Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning For Network Intrusion Detection. *IEEE Symposium on Security and Privacy (SP)*, 305–316.

13. Stouffer, K., Falco, J., & Scarfone, K. (2011). *Guide to Industrial Control Systems (ICS) Security (NIST SP 800-82 Rev. 2)*. National Institute of Standards and Technology.

14. Zhang, Y., & Chen, X. (2018). Deep learning for time series modeling and forecasting: A survey. *IEEE Transactions on Neural Networks and Learning Systems, 29*(11), 1–21.