# Physical AI Security: A Unified Five-Layer Taxonomy for a Systematic Survey from Silicon to Semantics

Ahmed M. Elmisery, Nabil Moukafih, Suga Junichi

**Abstract**

Collaborative robots, autonomous mobile robots (AMRs), and edge inference nodes inside Industrial Control Systems (ICS) now let AI models drive physical actuators. For these systems a cyberattack is no longer only a data problem: a successful exploit can disrupt infrastructure, halt production, or injure people. Research on Physical AI Security (PAIS) is split across hardware trust, artificial intelligence, robotics, operational technology, and cyber-physical systems, which makes it hard to see how risks and defenses depend on one another across the cycle of AI-driven sensing, decision-making, actuation, and feedback.

We propose a five-layer taxonomy for PAIS: hardware-rooted trust, pre-deployment validation, runtime assurance, domain semantics, and knowledge feedback. The taxonomy is organized around the physical decision and gives the fragmented literature a common structure in which cross-layer dependencies can be stated. Using it as the classification frame, we conduct a PRISMA-informed structured review of 312 primary papers published between 2015 and 2025 across six databases, supplemented by 28 standards, 14 regulatory frameworks, and 12 incident reports. Existing work covers individual layers well but does little to connect them across the Physical AI lifecycle. Coverage is thinnest exactly where Physical AI differs most from classical ICS: model provenance, validation of learned controllers, and human-robot interaction security. From these gaps we derive a prioritized agenda of ten research challenges and an unevaluated reference architecture with falsifiable evaluation criteria. The taxonomy is offered as a shared frame for organizing the field, assessing where existing approaches fall short, and directing work toward secure and resilient Physical AI systems.

**Keywords:** Physical AI; cyber-physical systems; robot security; operational technology; hardware attestation; runtime assurance; HRI security; adversarial machine learning; ICS security; edge AI; anomaly detection; formal verification; semantic integrity; behavioral digital twin.

# 1 Introduction

Between 2015 and 2025, learned models moved out of the data center and onto the factory floor. Stochastic models now *decide and act in the physical world*: they steer collaborative robot arms in manufacturing cells, route autonomous mobile robots (AMRs) through warehouses, add machine learning (ML) anomaly detection to programmable logic controllers (PLCs), infer control policies on edge nodes in power-grid substations, and run the humanoid service platforms now appearing in clinics and homes. In every one of these deployments the model's output reaches an actuator. We refer to them collectively as *Physical AI*: systems whose algorithmic decisions cause kinetic actuation, and whose failures can injure people, stop production, or take infrastructure offline [4, 5].

The transition is moving quickly, although the figures usually quoted for it come from market research rather than peer-reviewed measurement. ABI Research places the robotics market at approximately USD 50 billion in 2025, projected to reach USD 110.7 billion by 2030 at a 13.8% compound annual growth rate [121], and MarketsandMarkets sizes the OT security market at USD 27.39 billion in 2026, forecast to USD 58.94 billion by 2031 [122]. We report these as *vendor-reported market estimates*: they indicate order of magnitude and direction of travel, and no argument in this survey rests on them.

Incident telemetry gives a firmer picture of the security consequences. Dragos recorded 1,693 ransomware attacks against industrial organizations in 2024, an 87% year-on-year increase, with the number of ransomware groups observed targeting OT rising from 50 to 80 over the same period [119]; the following annual review recorded 3,300 affected industrial organizations and 119 tracked groups during 2025 [147]. These incident figures, not the market projections, are the baseline against which we assess Physical AI security.

As learned models replace deterministic logic and erode the old air gaps, the attack surface grows in four directions that conventional information systems (IS) governance does not cover. First, learned models bring their own attack classes: adversarial input perturbations, training-data poisoning, and model extraction have no counterpart in a deterministic rule-based controller [7, 8]. Second, edge inference now sits inside real-time plant loops, and the network separation that once kept operational technology out of reach of enterprise IT attacks has largely gone [9]. Third, humans and robots increasingly share a workspace, which opens a social-engineering channel in which an adversary exploits the trust an operator places in an autonomous machine [4]. Fourth, an attack can be spread across many commands. Each command is valid and passes invariant checks, but the sequence as a whole drives equipment toward wear-out or destruction, and a filter that checks one command at a time never sees it [10].

These four dimensions define the research problem of this paper. Securing Physical AI requires reasoning about a single physical decision from the silicon that computes it to the process it changes, and about how a weakness introduced at one point in that chain propagates to the others. We therefore review how existing research addresses this problem, organizing the literature around the five security dimensions that such reasoning requires.

*Hardware-rooted trust and AI supply-chain security.* Trusted execution environments and remote attestation are well characterized for general purpose platforms [13, 14], rest on a mature specification base [47], and are moving toward open, reviewable roots of trust [48, 49]. Hardware supply-chain compromise [71, 136], side-channel recovery of model weights [50, 51], and query-based model extraction [99] are each studied in depth. This literature, however, attests the *platform* rather than the *model*: binding a trained model, its provenance, and its runtime measurement to each inference output appears only implicitly in federated-learning work [15] and has not been formalized. It also rarely follows a compromise through to the physical consequence it enables downstream.

*Pre-deployment validation and AI assurance.* Formal verification of neural networks has progressed from exact SMT-based methods [23, 53] through reachability [22] and abstract interpretation [24] to bound propagation that runs on large networks [52, 158], with community benchmarking through VNN-COMP [55]. Specification-based monitoring provides the temporal logics in which cyber-physical properties can be stated [81, 103, 104], and security-oriented digital twins evaluate behavior against expected plant dynamics [17–19]. The limitations are structural. Certificates that remain tight at small scale loosen at the model sizes deployed in Physical AI; twins validate against invariants whose origin and integrity are assumed rather than established; and no invariant language spans robotics, OT, and HRI. Neither verification nor twin-based validation consumes evidence of *what* is being validated, nor hands an expected-behavior profile to the runtime monitor that follows.

*Runtime assurance and cyber-physical security.* This is the most mature dimension. Foundational CPS security work formalized false-data injection, process-aware attacks, and attack detectability [1, 3, 30, 36], and has been consolidated by several surveys [31–33]. ICS anomaly detection has grown around public testbeds [114–116], producing lightweight, one-class, and federated detectors [40, 42, 43]. Two weaknesses recur. Learned detectors are evadable by adversarially crafted inputs and sequences [7, 8, 10], and reported performance derives from a small number of process-control benchmarks whose attacks were scripted by hand. Runtime monitors also take their notion of "normal" as given; whether that baseline was itself validated or tampered with lies outside their scope.

*Human–robot and domain-semantic security.* Human-factors research has established that operators over-trust robots and defer to them under pressure [58, 59] and readily attribute intentions to robot behavior [60], but it treats over-trust as a usability and safety concern rather than as an attack surface. Haskard and Herath [4] give the strongest security treatment of this interface, within collaborative robotics. Language-grounded robot control [150] adds an instruction channel that is open to direct and indirect prompt injection [131, 149]. On the industrial side, process-aware attacks composed of individually valid commands [10, 36] motivate sequence-level semantic monitoring, which exists mainly as rule-learning and process-mining prototypes [20, 77]. HRI semantics and OT semantics are studied by different communities, and neither is connected to pre-deployment validation as a source of the invariants that validation needs.

*Knowledge feedback and continuous security adaptation.* Adversary knowledge bases catalogue attacks on ICS and on machine learning [124, 125]; tamper-evident evidence stores have been proposed using blockchains and signed logs [29, 73, 74]; LLM-assisted and learning-based methods automate red teaming and progressive hardening [12, 110, 162]. These contributions are typically framed as logging, compliance, or offline threat intelligence. The loop from an observed runtime deviation back to updated invariants and new pre-deployment test cases is rarely closed, and the knowledge store is almost never analyzed as a poisoning target in its own right.
*Combined treatments.* Several surveys combine two or more of these dimensions, and Table 1 positions this paper against the six nearest. CPS security surveys span runtime detection and OT semantics but predate learned control [31, 33]. Robotics security surveys add AI-enabled platforms but treat formal pre-deployment validation and hardware attestation as out of scope [5], or abstract away AI-specific attack surfaces such as model poisoning and digital-twin bypass [9]. The safety–trust–security treatment of collaborative robotics excludes industrial protocol layers, software provenance, and hardware roots of trust [4], and the agent-centric survey of AI in CPS provides no standards mapping [27]. Each is stronger than this paper on its own axis, and we cite them as such throughout.

Individual Physical AI components are not short of security mechanisms; each dimension above has an active literature. What the field lacks is a common structure that connects these mechanisms across the Physical AI lifecycle. The connection matters because vulnerabilities and assurance mechanisms at one layer influence, or depend upon, those at another: a validation twin that admits a decision is only as trustworthy as the attestation of the model that produced it and the integrity of the invariants it evaluates; a runtime monitor inherits its expected-behavior profile from validation; and a poisoned knowledge store silently degrades both. Without a shared structure, these dependencies cannot be stated precisely, the fragmented literature cannot be compared on common terms, and coverage gaps cannot be measured. Closing this gap requires a decision-centric taxonomy that defines the security role and boundary of each layer, states the interfaces between layers, and can be used as a classification frame for counting research effort. This paper proposes such a taxonomy, Physical AI Security (PAIS), outlined in Fig. 4 and defined in Section 4, and uses it as the analytical frame for a systematic review. We are explicit that the contribution is integrative rather than a first treatment: each constituent problem has been surveyed before, and the claim we defend is that no existing work spans hardware trust, pre-deployment validation, runtime assurance, HRI security, OT semantic integrity, and knowledge feedback in a single structure that permits cross-layer dependency and cascading-failure analysis.

The paper makes four contributions:

1. We formalize a decision-centric, five-layer security framework that follows a physical decision from silicon attestation to knowledge feedback, specifying the security role, boundary, and interfaces of each layer (Section 4), and use it to organize 19 representative attack types with strict evidence grading (Section 5).

**Table 1** Positioning against the nearest prior surveys. ✓=addressed substantively; Partial=addressed but not as a primary axis; ✗=out of declared scope. The contribution claimed here is *integrative*: every column has been covered before, but no prior work spans the full set.

| Survey | Primary scope | HW trust (L1) | Pre-dep. valid. (L2) | Run-time assur. (L3) | HRI security (L4a) | OT sem. (L4b) | Know. feed. (L5) | Frameworks / roadmap |
|---|---|---|---|---|---|---|---|---|
| **Humayed et al. 2017 [33]** | CPS security across four domains | ✗ | ✗ | Partial | ✗ | ✓ | ✗ | Domain taxonomy; no regulatory mapping |
| **Giraldo et al. 2018 [31]** | Physics-based attack detection | ✗ | Partial | ✓ | ✗ | ✓ | ✗ | Detection-method taxonomy only |
| **Neupane et al. 2023 [5]** | Security of AI-enabled robotics | ✗ | ✗ | ✓ | Partial | Partial | ✗ | No formal-methods or attestation axis |
| **Botta et al. 2023 [9]** | Robot cybersecurity | Partial | ✗ | ✓ | Partial | Partial | ✗ | No AI-specific attack surface |
| **Haskard & Herath 2025 [4]** | Safety–trust–security nexus in collaborative robotics | ✗ | Partial | ✓ | ✓ | ✗ | ✗ | Strongest prior HRI treatment; excludes OT and silicon |
| **Zhao et al. 2026 [27]** | AI agents in CPS; environmental interaction and deepfakes | ✗ | Partial | ✓ | ✓ | Partial | Partial | Agent-centric; no standards mapping |
| **This survey** | Physical AI security, silicon to semantics | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ATT&CK/ ATLAS/ D3FEND/ EMB3D + IEC 62443/ ISO 10218/ EU AI Act; 10-challenge roadmap |

The delta claimed is the *conjunction*: a decision-lifecycle layering that spans hardware trust through OT semantics, a corpus-based per-layer gap quantification, a combined MITRE and regulatory mapping, and a prioritized research agenda. Individually, several columns are better served by the works above, and we cite them as such throughout.

2. Using the taxonomy as the classification frame, we review 312 primary technical papers (2015–2025) with dual-coder inter-rater reliability ($\kappa \geq 0.71$) (Sections 3 and 6), backed by a sensitivity analysis that bounds the effect of the eligibility criteria on the coverage-gap findings (Section 9.2).
3. We show how the mechanisms identified across the layers connect along the decision lifecycle in an explicitly unevaluated reference architecture with contract-based seams (Section 7), and relate the taxonomy to five MITRE knowledge bases, established security frameworks, and ten standards and regulatory instruments,

including IEC 62443, ISO 10218-1/2:2025, and the staged obligations of the EU AI Act (Section 8).
4. We derive ten research challenges (R1–R8, R2b, R4b) from the measured gaps, each with its open problem and enabling technologies, together with falsifiable evaluation criteria for the reference architecture (Section 9).

The remainder of this paper is organized as follows. Section 2 defines Physical AI, presents a generic Physical AI configuration, and fixes the scope of the taxonomy and review. Section 3 describes the review methodology and the role of standards, regulations, and incident reports. Section 4 presents the five-layer PAIS taxonomy, and Section 5 uses it to organize the threats to Physical AI. Section 6 provides the layer-by-layer systematic literature analysis. Section 7 connects the surveyed mechanisms in an end-to-end reference architecture. Section 8 maps the taxonomy to security knowledge bases, frameworks, standards, and regulations. Section 9 sets out open research challenges, Section 10 discusses cross-layer security implications, and Section 11 concludes the paper.

# 2 Physical AI: Definition, Scope, and Security Context

## 2.1 Definition and Conceptual Positioning

"Physical AI" is a recent label for an intersection of several older research traditions. We define *Physical AI* as any system in which an AI model (a deep neural network, a reinforcement learning agent, or another learned controller) produces outputs causally coupled to actuators that can apply forces, move matter, or change energy flows in the physical world.

The definition overlaps with three neighboring terms. The literature on *AI-enabled cyber-physical systems (CPS)* has concentrated on control-theoretic stability and physics-based anomaly detection over networked relay or deterministic logic [31, 33]; Physical AI is the subset of CPS in which stochastic, learned models replace or augment the deterministic controller. *Embodied AI*, as the term is used in computer vision and robotics, is about sensorimotor learning, spatial reasoning, and sim-to-real transfer; Physical AI inherits these perception–action models and adds security attestation, industrial OT semantics, and safety-critical failure modes to the scope. Research on *smart robotics and autonomous systems* has usually treated embedded devices, motion planning, and communication links as separate problems, whereas Physical AI security follows the whole decision lifecycle, from hardware roots of trust through pre-deployment verification and edge runtime assurance to the semantics of human interaction [4, 5, 9].

This breadth distinguishes Physical AI security from both traditional IT security and classical OT security. Unlike IT, the consequence of a breach is kinetic; unlike OT, the threat includes adversarial AI techniques (model poisoning, adversarial examples) with no direct analogue in relay-logic controllers.

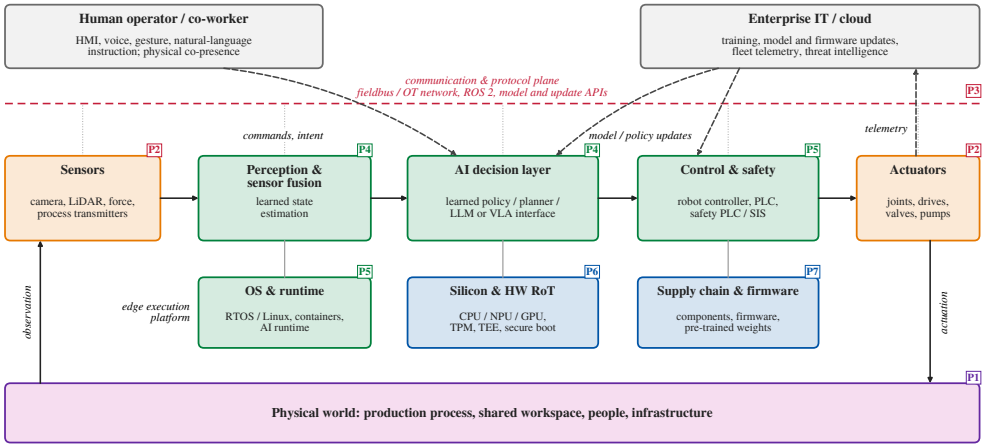



**Fig. 1** Generic Physical AI configuration used to bound the PAIS taxonomy and the literature review. A closed sense–decide–act loop (middle row) couples a learned decision layer to actuators acting on the physical world (bottom). The learned components execute on an edge platform (lower row) whose trust depends on silicon and the supply chain, and the loop is connected to a human operator and to enterprise or cloud services over a shared communication and protocol plane. Dashed arrows denote supervisory or asynchronous couplings. Tags P1–P7 give the attack plane of Fig. 2 on which each component sits.

## 2.2 A Generic Physical AI Configuration

Fig. 1 abstracts the deployments above into a generic Physical AI configuration. A closed sense–decide–act loop links the physical world to sensors, learned perception and sensor fusion, an *AI decision layer* (a learned policy, a planner, or a language- or vision-language-conditioned interface), a control and safety layer (robot controller, PLC, or safety instrumented system), and actuators that act back on the physical world. The learned components execute on an edge platform comprising an operating system and AI runtime, silicon with a hardware root of trust, and firmware, components, and pre-trained weights obtained through a supply chain. A communication and protocol plane connects the loop's components to one another, to a human operator who issues commands and may share the workspace, and to enterprise or cloud services that supply model and policy updates and receive telemetry.

Two properties of this configuration shape the rest of the paper. First, the AI decision layer sits between perception and control, so its output is causally coupled to actuation, frequently with only deterministic safety logic in between. Second, trust in any single decision depends on elements outside the loop itself: the platform that executed the model, the supply chain that delivered it, the human who instructed it, and the updates that changed it. The taxonomy of Section 4 is defined over the components of Fig. 1, and each component is tagged with the attack plane of Fig. 2 on which it sits.

## 2.3 Scope Boundaries for the Taxonomy and Review

The definition and configuration above fix the boundaries that the taxonomy and the literature review inherit. Four boundary dimensions are summarized in Table 2: the systems covered, the architectural elements analyzed, the security concerns considered, and the operational contexts from which evidence is drawn.

Physical AI security is therefore concerned with systems whose failures can cause bodily harm, production disruption, or the loss of critical infrastructure [4, 5]. Chat-only language models and purely virtual agents fall outside it. Six system classes fall inside:

- *Industrial robots and cobots*: manipulator arms for manufacturing, welding, assembly, and logistics, operating under ISO 10218-1/2:2025, which now absorbs the collaborative-application content formerly in ISO/TS 15066 [153, 154].
- *AMRs and AGVs*: sensor-navigated platforms in warehousing and logistics.
- *OT controllers*: PLCs, distributed control systems (DCS), and remote terminal units (RTUs) with ML-based predictive control or anomaly detection.
- *SCADA systems with AI components*: supervisory layers that use ML for demand forecasting, fault prediction, or intrusion detection.
- *Edge AI inference nodes*: embedded compute (NVIDIA Jetson, Raspberry Pi with TPM, ARM Cortex-M with TrustZone) running neural network inference whose outputs feed physical control loops.
- *Humanoid and service platforms*: embodied platforms now being trialled in clinical, domestic, and defense settings.

These boundaries determine where a PAIS layer starts and stops (Section 4) and which records are eligible for the systematic review (Section 3).

## 2.4 The Cyber-Physical Attack Surface

IT security models generally assume that a successful attack stays digital: data is exfiltrated, a service is denied, privileges are escalated. The assumption fails once learned models drive actuators. Fig. 2 divides the resulting attack surface into seven planes, running from the silicon to the environment in which the actuators move.

The planes show where Physical AI departs from enterprise IT and from classical operational technology (OT). The two hardware planes, Supply Chain & Firmware (Plane 7) and Silicon & Hardware RoT (Plane 6), are conceptually the same as in IT hardware security, but here their failure has physical consequences: once the root of trust is compromised, downstream validation can no longer be relied on, and an attacker can substitute a model without being detected.

Above them, the OS & Runtime Plane (Plane 5) provides execution isolation, and the AI Model & Inference Plane (Plane 4) carries the learned vulnerabilities specific to AI-enabled systems; adversarial perturbations at Plane 4 have no counterpart in deterministic relay-logic OT. The Communication & Protocol Plane (Plane 3) is the most thoroughly studied plane in classical industrial control system (ICS) security [32, 45], but in Physical AI it also carries model API traffic, remote weight downloads,

**Table 2** Scope boundaries of the PAIS taxonomy and systematic review

| Dimension | In scope | Boundary / out of scope |
|---|---|---|
| **Systems** | Industrial robots and cobots; AMRs and AGVs; ML-augmented PLC, DCS, RTU, and SCADA; edge AI inference nodes; humanoid and service platforms | Chat-only language models and purely virtual agents with no actuation path |
| **Architecture** | All components of Fig. 1: sense–decide–act loop, edge execution platform and its supply chain, communication plane, human operator, update and telemetry interfaces | Enterprise IT and cloud are analyzed as entry points and update sources, not as protection targets in their own right |
| **Security concerns** | Integrity, availability, and safety of physical outcomes; authenticity of human commands and model provenance; confidentiality where it enables a physical-consequence attack (e.g., model extraction) | Informational harms of AI outputs with no path to actuation; privacy treated only where it enables or constrains physical-consequence attacks |
| **Operational contexts** | Manufacturing, logistics and warehousing, energy, water, and process industries; shared human–robot workspaces in industrial, clinical, and domestic settings | Autonomous road vehicles meet the definition but have a dedicated literature and regulatory regime; they are used as sources of transferable attack evidence, not as a target platform class |

and federated-learning orchestration. The Actuator & Sensor Plane (Plane 2) and the Physical World Plane (Plane 1) form the kinetic boundary. Plane 1 exists in legacy OT security as well. What changes is that an adversary who understands the weaknesses of a learned sensor-fusion stack can alter the physical environment to steer the model's inference, and through it the physical outcome.

Attack chains in Physical AI usually cross several planes. A malicious firmware modification in the supply chain (Plane 7) can alter model behavior at Plane 4, which in turn produces an anomalous actuation sequence at Plane 2 and physical damage at Plane 1. A defense confined to one plane cannot see such a chain from end to end. The multi-plane view extends earlier CPS security work [1, 3, 30, 33].

AI-specific exploits also differ from classical ICS attacks in speed and detectability. The best-known ICS campaigns unfolded over weeks, whereas an adversarial perturbation against a perception loop can cause a collision within milliseconds. Classical control attacks usually leave process anomalies that show up within minutes; a white-box adversarial example can keep an asset's outputs invariant-compliant until the moment the harmful action executes. How to combine physics-aware verification [31] with non-linear learned controllers remains an open question.

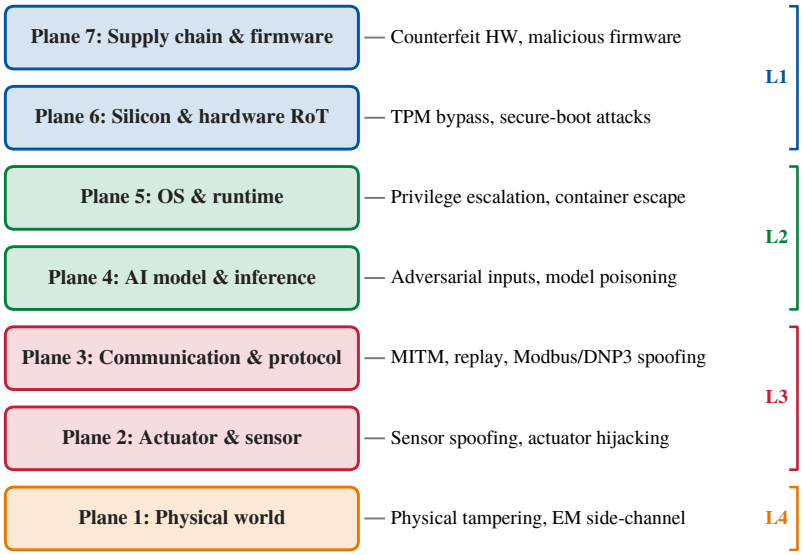



**Fig. 2** The seven-plane Physical AI attack surface. Planes are numbered outward from the kinetic environment in which the system acts (Plane 1, physical world) to the origin of its components (Plane 7, supply chain and firmware); the narrative order used in Section 4 traverses the same planes in the opposite direction, from silicon upward. The alignment between these threat vectors and the five defense layers is detailed in Section 4.

**Table 3** Reference ICS security incidents

| Incident | Attack vector | Physical consequence |
|---|---|---|
| **Stuxnet (2010)** | PLC firmware modification via USB | Centrifuge destruction [35] |
| **Ukraine Grid (2015/16)** | SCADA credential theft + Industroyer | 225,000 homes lost power |
| **Oldsmar Water (2021)**† | Remote HMI access via TeamViewer (as initially reported) | Sodium hydroxide set-point raised |
| **Triton/TRISIS (2017)** | Safety-system controller reprogramming | Production shutdown, safety system disabled |
| **Colonial Pipeline (2021)** | IT ransomware propagating to OT | 5,500-mile pipeline shutdown |

†Attribution disputed: the FBI later stated it could not confirm that the event was initiated by a targeted intrusion [164].

## 2.5 Reference Incidents

Table 3 lists the incidents we use as reference points. None of them involved a learned controller, but together they show how attacks move between planes. Stuxnet (2010) is the canonical multi-plane attack: USB delivery at Plane 7, PLC firmware modification at Planes 6–5, and centrifuge frequency manipulation at Planes 2–1 [35]. Triton/TRISIS (2017) went after safety instrumented systems (SIS), which showed

that adversaries are prepared to attack the safety mechanisms themselves. The Oldsmar incident (2021), as first reported, illustrates the HMI attack surface: a remote session raised a chemical set-point through valid HMI commands, which an AI-mediated HMI without semantic checks would not have caught. Its attribution is disputed (the FBI later stated that it could not confirm a targeted intrusion [164]), so we use it only to illustrate the attack surface, not as a confirmed attack. Colonial Pipeline (2021) was an IT-to-OT spillover of a different kind. Ransomware on the IT network led operators to shut down OT systems as a precaution, so the organization's own response caused the physical disruption.

# 3 Systematic Review Methodology

We conducted a structured literature review and followed the PRISMA 2020 reporting guidelines for search documentation and exclusion tracking [123]. We searched IEEE Xplore, ACM Digital Library, arXiv, Scopus, Web of Science, and Semantic Scholar using Boolean queries combining primary infrastructure terms (*Physical AI, cyber-physical systems, industrial control systems, robot security, operational technology*) with secondary technical terms (*adversarial ML, attestation, digital twin, runtime assurance, HRI security, ICS anomaly detection, SCADA security, edge AI trust, OT semantic integrity, hardware root of trust, formal verification of neural networks*).

The screening window for the quantitative corpus was January 2015 to December 2025 inclusive. A small number of sources published between January and July 2026 are also cited, mainly threat-intelligence annual reports, 2026 preprints on LLM-mediated control, and the 2026 amendment of the EU AI Act. These are *kept out of every quantitative claim*: they are excluded from the 312-paper corpus, from all layer assignments and counts, and from the coverage statistics in Figs. 6 and 18. They are cited only as narrative evidence of emerging trends.

Full PRISMA 2020 protocol compliance (prospective registration, specialized data extraction forms, within-study risk-of-bias assessment) was not achieved; this constraint is documented in the Limitations section.

## 3.1 Search Flow and Eligibility Criteria

The primary search retrieved 3,200 records across target indexes. Of these, 1,847 were excluded at title/abstract screening (off-topic, non-English, duplicate), leaving 1,353 for full-text review. A further 1,060 were excluded for lacking a primary technical contribution (631), being a review/editorial without novel method (203), having fewer than five citations when published pre-2024 (148), or being a standards document separated into the compliance corpus (78). This yields 293 papers from the primary search. The supplementary HRI-targeted search described below screened a further 214 records and contributed 19 papers.

*Corpus accounting:* The headline figure of 312 primary papers includes the 19 supplementary HRI papers ($293 + 19 = 312$), and 3,414 records were screened across both rounds ($3{,}200 + 214$). Every layer count in this survey, including the L4a total of 57, is computed on the 312-paper corpus after the supplement. We also reviewed 28

international standards documents, 14 regulatory frameworks, and 12 major incident reports as separate corpora (Fig. 3).

## 3.2 HRI Supplementary Search

To address the risk that "HRI security" as the sole query term under-captures adjacent literature, we conducted a supplementary round using alternate terms (*robot manipulation attack, operator deception robot, authority spoofing cobot, human-robot trust exploitation, cobot social engineering*). This round screened 214 additional records, of which 19 passed full-text review and entered the L4a corpus, bringing the L4a total to 57 papers. The sparsity finding therefore does not depend on the search string: a targeted second round added 19 papers to the layer and L4a remained the least-covered layer by a wide margin.

## 3.3 Screening, Classification, and Inter-Rater Reliability

Title/abstract screening and full-text eligibility assessment were performed independently by the first and second authors (A1, A2), each blind to the other's decisions, with disagreements reconciled by discussion. Layer classification of all 312 primary papers was likewise performed independently and blind by A1 and A2. Disagreements were adjudicated by the third and fourth authors (A3, A4), who took no part in first-pass coding.

Cohen's $\kappa$ was computed between the A1 and A2 *first-pass* label sets on a 20% random sample ($n = 63$ papers). Per-layer inter-rater reliability is: L1 (Trust & Execution) $\kappa = 0.81$; L2 (Pre-deploy. Validation) $\kappa = 0.76$; L3 (Runtime Assurance) $\kappa = 0.84$; L4a (HRI Semantics) $\kappa = 0.71$; L4b (OT Semantics) $\kappa = 0.79$; L5 (Knowledge Feedback) $\kappa = 0.73$. All exceed the $\kappa \geq 0.61$ band characterizing substantial agreement [2].

*Representative search string (IEEE Xplore):* `(“Physical AI” OR “cyber-physical” OR “ICS security” OR “robot security” OR “OT security”) AND (“adversarial” OR “attestation” OR “digital twin” OR “anomaly detection” OR “HRI security” OR “semantic integrity” OR “formal verification”)`. The query was adapted to the syntax of each database.

## 3.4 Standards, Regulatory, and Incident Corpora

Three further corpora were reviewed separately from the primary papers because they carry different evidentiary weight and are used for different purposes. Standards documents retrieved by the primary search were removed from the primary corpus at full-text review (Section 3.1) rather than counted as technical contributions.

*Standards and regulatory frameworks* (28 standards documents and 14 regulatory frameworks) comprise instruments that impose requirements on at least one system class within the scope of Table 2. They are the basis of the mapping in Section 8: each instrument is read at clause level against the security role of each PAIS layer (Section 4) and scored as none, partial, or addressed. Ten of these instruments are charted in the compliance heatmap of Section 8.

*Incident reports* (12) ground the threat taxonomy. They supply the real-world attack chains summarized in Section 2.5 and anchor the *analogous-evidence* grade of the threat taxonomy (Section 5): a threat demonstrated only against deterministic control logic, as in the canonical ICS incidents, is graded as analogous rather than as demonstrated on Physical AI.
*Grey literature* (vendor threat intelligence and market reports) is retained only where it is the sole source of operational incident data, a common condition in OT security, and is weighted below peer-reviewed and incident-response sources throughout. No gap finding, layer count, or research-challenge prioritization in this paper rests on a vendor-reported statistic, and sources dated 2026 are excluded from all counts as described above.

## 3.5 Data Extraction and Quantitative Synthesis

The PAIS taxonomy (Section 4) is the classification frame for the quantitative synthesis. Each primary paper was assigned to every layer whose security role its primary technical contribution addresses, using the layer definitions and boundaries of Table 4. A paper can therefore carry more than one label, and the 312 papers yield 620 paper–layer assignments across L1–L4b. L5 labels were assigned and reliability-tested in the same way (Section 3.3); because knowledge-feedback contributions are cross-cutting, L5 is analyzed qualitatively (Section 6.6) rather than counted alongside the other layers.

For each paper we also recorded the publication year, the sub-topic within its layer (for example, the three L1 sub-topics of Fig. 6), and, where reported, the evaluation platform or benchmark and the detection metrics (F1, false-positive rate, latency) summarized in Tables 10 and 14. Layer counts and publication trends are computed over the 312-paper corpus only. Performance figures are reported as min–max ranges across the corpus studies that report each method; they are not pooled estimates or confidence intervals, and no meta-analysis is attempted. The robustness of the coverage-gap findings to the citation-based eligibility criterion is examined in Section 9.2.

## 3.6 Scope and Limits of the Evidence Base

Because this survey reports quantitative detection figures and a gap analysis, we state the limits of its evidence base here, before any synthesis, rather than only in the Limitations section. Three constraints bound every empirical claim that follows.

First, *benchmark provenance*: all detection-performance figures synthesized in this survey derive from three public ICS benchmarks (SWaT, BATADAL, HAI) covering water treatment, water distribution, and steam-turbine process control. Their attacks were manually scripted by researchers, not generated adversarially. Reported figures are therefore upper bounds under idealized, rule-bound threat conditions.

Second, *platform transferability*: those three benchmarks are process control plants. The scope named in this survey's title, Physical AI, includes cobots, AMRs, humanoid platforms, and LLM-mediated robots, for which no equivalent public evaluation exists. Detection results should not be transferred to those platforms, and we do not transfer them.

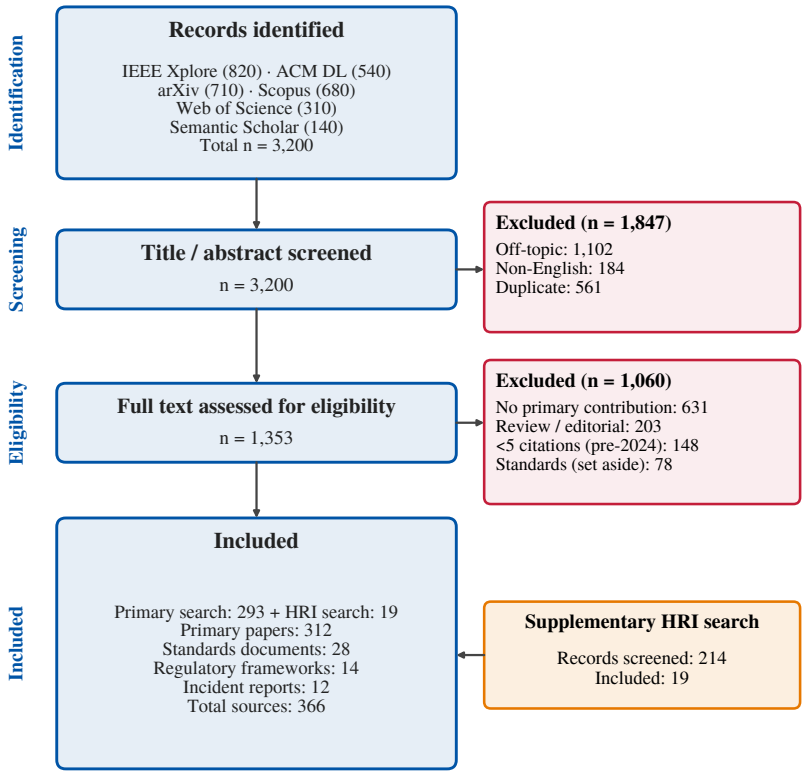



**Fig. 3** Structured literature review flow (PRISMA 2020 reporting guidelines followed for search documentation and exclusion categories). Primary search: 3,200 records identified across six databases, yielding 293 papers. A supplementary HRI-targeted search screened a further 214 records and contributed 19 papers, giving 3,414 records screened in total and a primary corpus of 312 papers.

Third, *HRI evidence status*: no public benchmark contains HRI-specific attacks. Every performance claim in Section 6.4 is theoretical or extrapolated from an adjacent domain, and Table 11 grades each attack class accordingly.

The survey is therefore representative of *the literature on Physical AI security* and of *OT anomaly-detection research in particular*. Section 9.4 develops these constraints and Research Challenge R7 addresses them.

# 4 The Proposed Unified Five-Layer PAIS Taxonomy

## 4.1 Design Rationale

We propose a *five-layer Physical AI Security (PAIS) taxonomy* built on two things: the lifecycle of a single physical AI decision, and the separation of concerns that safe, secure, and auditable systems need. Three design properties set it apart from earlier frameworks such as the MITRE ATT&CK ICS matrix, the Purdue Model, and existing CPS security surveys [31, 33]:

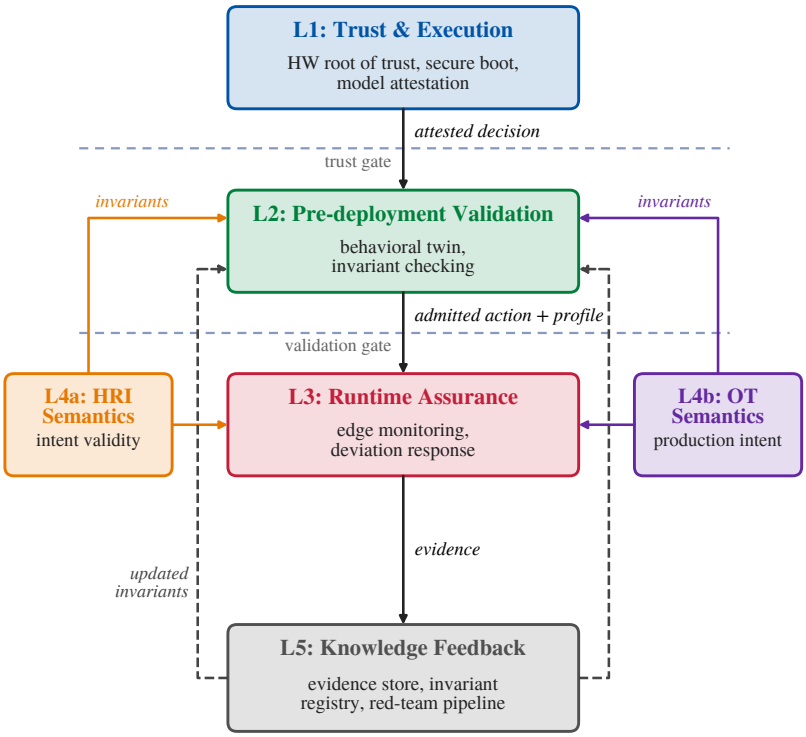



**Fig. 4** The five-layer Physical AI Security (PAIS) taxonomy. Solid arrows show the real-time decision path; dashed arrows show the asynchronous knowledge-feedback path.

*(1) Decision-centric.* The taxonomy organizes around the lifecycle of a *physical decision*, from attestation of origin, through pre-deployment validation and runtime monitoring, to feedback of evidence back into the security posture. It follows how Physical AI actually works, not how IT/OT architectures have traditionally been partitioned.
*(2) AI-native.* Traditional taxonomies treat software layers as undifferentiated. PAIS has an explicit *trust layer* (L1) for model integrity, model attestation, and supply-chain provenance, which the ICS and robotics security frameworks we reviewed do not cover, and it targets the adversarial ML threat surface documented by MITRE ATLAS [124].
*(3) Cross-domain.* The taxonomy integrates robotics (HRI semantics, L4a), industrial control (OT semantics, L4b), and edge inference (runtime assurance, L3) in one structure, which is what makes the cross-layer dependency analysis of Section 7 possible.

The three properties follow from a single design principle: every AI-driven physical decision must be (1) trusted at its origin, (2) validated before it reaches the actuator, (3) monitored while it executes, and (4) folded back into shared knowledge once it completes. The taxonomy turns this principle into five layers, with a published contract at each seam between them. Fig. 4 gives an overview of the resulting structure.

## 4.2 Layer Definitions, Security Roles, and Boundaries

Each layer is defined by the question it answers about a physical decision, the security role it plays, what it consumes from and produces for the other layers, and what it deliberately does *not* do. Table 4 summarizes these elements, and the subsections below describe each layer.

### 4.2.1 Layer 1: Trust and Execution

Layer 1 establishes that a Physical AI decision originates from an *uncompromised, measurable stack*. Its question is: "Who signed this decision, and can we verify the entire chain from silicon to inference?" Mechanisms include hardware roots of trust (HRoT), TPM 2.0 [47], ARM TrustZone / AMD SEV / Intel SGX [13, 14], secure boot, remote attestation, model integrity hashing, and supply-chain provenance. L1 does *not* judge whether the decision is physically safe; that is L2's job.

### 4.2.2 Layer 2: Pre-deploy. Validation

Layer 2 decides whether an *admitted, attested* decision is physically safe before it is dispatched to an actuator. Its question is: "Given what the system is trying to do, will it violate any safety, security, or operational invariant?" Mechanisms include behavioral digital twins [17, 19], formal verification of invariants [22, 23], and model-checking of cyber-physical properties. L2 does not own the invariants; it takes them from L4 (domain semantics) and L5 (the registry).

### 4.2.3 Layer 3: Runtime Assurance

Layer 3 continuously monitors whether the physical system *executes as validated* and responds locally to deviations. Its question is: "Is the physical world doing what was admitted, and can we contain deviations before they escalate?" Mechanisms include edge-deployed anomaly detectors [7, 8], telemetry comparison against expected-behavior profiles, and local containment actions (slow-stop, safe-state, hand-back-to-human). L3 does not re-validate decisions; it only checks that they are executed faithfully.

### 4.2.4 Layer 4: Domain Semantics

Layer 4 translates domain knowledge (what humans mean, what a production line is trying to achieve) into invariants and intent claims that L2 and L3 can act on. It has two peer sub-layers.

*L4a, HRI semantics,* validates human intent, resists trust-manipulation attacks (impersonation, social engineering of robots), and supplies HRI-specific invariants such as proximity envelopes, gesture-meaning mappings, and authority hierarchies [4].

*L4b, OT semantics,* models production intent in manufacturing flows and detects when individually valid OT commands compose into semantically incorrect outcomes [10]. Neither sub-layer gates or contains actions directly; they supply semantics to L2 and L3.

**Table 4** PAIS layer definitions: guiding question, security role, interfaces, and boundary. The "does not" column states the boundary used to decide layer membership during classification (Section 4.4).

| Layer | Guiding question | Security role | Consumes → produces | Does not | Planes |
|---|---|---|---|---|---|
| **L1 Trust & Execution** | Who produced this decision, and can the chain from silicon to inference be verified? | Establish integrity and provenance of platform, firmware, and model | Measurements, keys → attested decision envelope (to L2); signed attestation records (to L5) | Judge whether a decision is physically safe | P7, P6 |
| **L2 Pre-deploy. Validation** | Will the attested decision violate a safety, security, or operational invariant? | Admit, constrain, or deny a decision before actuation | Attested decision (L1), invariants (L4, L5) → verdict and expected-behavior profile (to L3); validation reports (to L5) | Own the invariants; monitor execution | P5, P4 |
| **L3 Runtime Assurance** | Is the system executing what was admitted, and can deviations be contained? | Monitor execution against the admitted profile; graded local containment | Expected profile (L2), invariants (L4) → containment actions; deviation events (to L5) | Re-validate decisions | P3, P2 |
| **L4a HRI Semantics** | Is this command a genuine, uncoerced expression of an authorized human's intent? | Supply intent claims and HRI invariants (proximity, authority, gesture meaning) | Human commands and context → intent claims and invariants (to L2, L3); HRI anomaly logs (to L5) | Gate or contain actions directly | P1 (human) |
| **L4b OT Semantics** | Does this sequence of individually valid commands serve the production intent? | Supply production-intent invariants; detect harmful command compositions | Process model, command traces → invariants (to L2, L3); OT violation events (to L5) | Gate or contain actions directly | P1 (process) |
| **L5 Knowledge & Feedback** | What has the system learned, and can that knowledge be trusted? | Persist evidence; maintain the invariant registry; generate red-team scenarios | Records from L1–L4 → updated invariants (to L2, L3); new test cases (to L2) | Sit on the real-time decision path | Cross-cutting |

### 4.2.5 Layer 5: Knowledge and Feedback

Layer 5 collects structured evidence from the other layers (signed validation reports, deviation events, attestation records), keeps it in a tamper-evident store, and feeds it back into invariant registries and red-team scenario pipelines [29]. It closes the learning loop without sitting on the real-time decision path: latency-sensitive decisions never wait on L5.

## 4.3 Relationships Between Layers

The layers are related in three ways. First, along the *real-time decision path* (solid arrows in Fig. 4), a decision passes a trust gate (L1 to L2) and a validation gate (L2 to L3) before and during actuation; each gate consumes an artifact produced by the layer above it, the attested decision envelope and the validation report with its expected-behavior profile respectively. Second, L4a and L4b are *semantic suppliers* rather than gates: they furnish the intent claims and invariants that L2 validates against and L3 monitors against. Third, L5 closes an *asynchronous feedback path* (dashed arrows): it receives evidence from every other layer and returns updated invariants and new test cases, but no latency-sensitive decision waits on it. These relationships are the seams at which cross-layer dependencies, and therefore cascading failures, arise; Section 7 makes them explicit as contracts.

The layers also map onto the attack surface. The seven planes of Fig. 2 are finer-grained than the defensive functions needed to protect them, so each layer groups adjacent planes into one security boundary instead of forcing a one-to-one mapping:

1. *Layer 1 ↔ Planes 7 and 6.* L1 covers the Supply Chain & Firmware Plane and the Silicon & Hardware RoT Plane. It relies on hardware roots of trust (HRoT), secure boot chains, and remote attestation to establish that downstream inference runs on an uncompromised software stack.
2. *Layer 2 ↔ Planes 5 and 4.* L2 covers the OS & Runtime Plane and the AI Model & Inference Plane. It intercepts attested decisions and checks each proposed action profile against cyber-physical invariants, using behavioral digital twins and formal verification, before any command is dispatched to an actuator.
3. *Layer 3 ↔ Planes 3 and 2.* L3 monitors traffic and hardware state on the Communication & Protocol Plane and the Actuator & Sensor Plane. Edge-resident anomaly detectors with sub-50 ms latency compare live telemetry with the expected-behavior profile, so kinetic deviations can be contained locally without waiting for cloud coordination.
4. *Layer 4 ↔ Plane 1.* L4 addresses attacks on the Physical World Plane by turning operational reality into context that a machine can check. L4a models the boundaries of operator interaction so that operator trust is harder to manipulate; L4b models production intent so that multi-step command sequences leading to wear-out or process failure can be caught.
5. *Layer 5 ↔ all planes.* L5 does not watch a single entry vector. It is asynchronous, cross-cutting infrastructure that collects signed attestation logs, validation reports, and deviation events from all seven planes, keeps them in a

tamper-evident store, and uses them to update the invariant registries and drive automated red-team scenario generation.

### 4.4 The Taxonomy as a Classification Frame

The taxonomy is also the common analytical frame used to classify and synthesize the literature in Sections 5–8. Four rules make that use reproducible. (i) *Guiding-question membership:* a paper, threat, or control is assigned to every layer whose guiding question (Table 4) its primary contribution answers, so assignments are multi-label. (ii) *Boundary exclusion:* a contribution is not assigned to a layer on the strength of a function that Table 4 lists under that layer's "does not" column; a platform-attestation scheme, for instance, is not a validation contribution merely because an attestation failure blocks actuation. (iii) *Defensive control point:* threats and adversary tactics are assigned to the layer at which the corresponding defense sits, not to the attacker's objective. (iv) *Semantic domain:* Layer 4 contributions are assigned to L4a when the semantics concern human intent or authority and to L4b when they concern production or process intent. Section 3.5 describes how these rules were applied to the 312-paper corpus, and Section 6 reports the resulting layer-by-layer analysis.

## 5 Threat Taxonomy for Physical AI

### 5.1 Threat Classification Framework

This section applies the five-layer structure of Section 4 to the threats facing Physical AI. Representative attacks are identified from the primary corpus and the incident corpus and organized by the PAIS layer at which the corresponding defense sits (classification rule (iii), Section 4.4). We classify Physical AI threats along three orthogonal axes: (1) *attack plane* (hardware, software, network, social); (2) *attack target* (trust layer, model, communication, actuator, human operator); (3) *physical consequence class* (safety, availability, integrity, production quality, audit integrity). All three axes are needed because Physical AI attacks often span planes: a supply-chain compromise at the hardware plane enables model substitution at the software plane, which then causes actuator misbehavior at the physical plane. Table 5 maps 19 representative threat types across these dimensions and, following the evidence-grading scheme used for the HRI taxonomy in Table 11, records for each whether it has been demonstrated on a Physical AI platform, is supported only by analogous evidence from adjacent non-AI systems, or remains theorized.

Read by layer, Table 5 shows where the threat evidence is concentrated. L1 has the largest group of threat types (six), but only model poisoning and side-channel leakage have been demonstrated against learned control. L3 holds both demonstrated attacks on learned perception (adversarial sensor input and sensor spoofing). Every L4a and L5 threat type is graded analogous or theorized: the layers the corpus covers least (Section 9) are also those whose threats rest on the weakest evidence. The subsections below examine the attack classes that cut across layers.

**Table 5** Physical AI threat taxonomy: 19 representative attack types organized by PAIS layer (groups: L1, L2, L3, L4a, L4b, L5, all layers) and mapped to attack planes and physical consequence classes. The Trusted Insider Abuse row spans all layers; an insider with valid credentials defeats L1–L4 authentication simultaneously.

| **Attack Name** | Layer | Plane | Technique | Example | Consequence | Ev. |
|---|---|---|---|---|---|---|
| **Firmware tampering** | L1 | Hardware/SW | Supply-chain injection | Counterfeit PLC firmware | Integrity | A |
| **Secure boot bypass** | L1 | Hardware | Side-channel, glitch | JTAG debug exploit | Trust subversion | A |
| **Supply chain compromise** | L1 | Supply chain | Counterfeit components | Malicious HSM | Trust subversion | A |
| **Side-channel leakage** | L1 | Physical | Power analysis, EM | Model weights extraction | IP/confidentiality | D |
| **Model extraction** | L1 | Network | Query-based extraction | Stolen inference model | IP theft / backdoor | A |
| **Model poisoning** | L1/L2 | Software | Training data manipulation | Backdoored CNN model | Actuation corruption | D |
| **Digital twin poisoning** | L2 | Software | Model inversion | Manipulating DT ground truth | Validation bypass | T |
| **Stealthy wear-out attack** | L2/L3 | AI-assisted | DRL-crafted sequence | Quality-degrading ICS commands | Production quality | D |
| **Adversarial sensor input** | L3 | Physical/Net | Adversarial examples | Lidar spoofing for AMR | Safety | D |
| **Sensor spoofing** | L3 | Physical/EM | GPS/LIDAR/GNSS spoof | Robot navigational error | Safety | D |
| **MITM on control channel** | L3 | Network | Replay/injection | Modbus TCP replay | Availability | A |
| **Ransomware in OT** | L3/L5 | Software | Lateral movement | Colonial Pipeline-class | Availability | A |
| **Multi-stage APT on ICS** | L3/L4b | Multi-plane | Industroyer-class | Triton/TRISIS-class$^\star$ | Safety / Availability | A |
| **Intent spoofing (HRI)** | L4a | Social/Net | Deepfake commands | Fake voice command to cobot | Safety | T |
| **Authority escalation (HRI)** | L4a | Social | Social engineering | Impersonating supervisor | Trust subversion | A |
| **Semantic OT attack** | L4b | Protocol | Command composition | Valid individual cmds $\rightarrow$ wrong outcome | Production quality | A |
| **Evidence store tampering** | L5 | Software | Log manipulation | Erasing anomaly records | Audit integrity | T |
| **Invariant poisoning** | L5 | Software | Registry manipulation | Weakening safety bounds | Validation bypass | T |
| **Trusted Insider Abuse** | All layers | Physical/ Social | Authority + access | Operator-as-attacker | All consequence classes | A |

$^\star$Triton/TRISIS targeted SIS controllers (L3 safety layer); listed here for ICS APT class coverage.
Trusted Insider Abuse bypasses L1–L4 authentication; mitigated only by dual-operator consensus (L4a) and audit logs (L5).
**Evidence grade** (same scheme as Table 11): **D** = demonstrated against a system in which a learned model drives physical actuation; **A** = analogous evidence only, i.e. demonstrated against an ICS, IT, or non-AI robotic system whose control logic was not learned; **T** = theorized, with no public demonstration in any setting. The grading is deliberately strict: the canonical ICS incidents (Stuxnet-class semantic attacks, Industroyer/TRISIS, Colonial Pipeline) grade **A** rather than **D** because the compromised control logic was deterministic, not learned. The resulting distribution (5 D, 10 A, 4 T) is a finding in its own right: for 14 of the 19 threat types listed here, the Physical AI community is reasoning from adjacent-domain evidence or from theory alone. Research Challenge R7 (benchmarks and testbeds) targets this gap.

## 5.2 Cross-Layer Threat: The Trusted Insider

Many Physical AI security discussions leave out the *trusted insider*. A human operator with legitimate credentials, physical access, and knowledge of the Physical AI deployment can simultaneously defeat L1 (presenting valid attestations), L2 (supplying correct invariant-compliant commands), L3 (there is no anomaly: the commands are what this operator normally sends), and L4a (biometric liveness passes). The only architectural defenses are dual-operator consensus for high-consequence actions (L4a) and tamper- evident audit logs (L5) that allow post-hoc forensic detection. An external attacker has to get through the PAIS layers one after another; an insider is already inside all of them. Physical access control, personnel security, and separation of duties therefore remain necessary alongside the technical architecture.

## 5.3 Adversarial Machine Learning in Physical AI

Adversarial ML attacks are especially dangerous in Physical AI because a perturbation too small to see in pixel or latent space can produce a large physical displacement. Biggio and Roli [117] and Papernot et al. [118] formalized the standard adversarial ML threat model in terms of the attacker's goals, knowledge, and capabilities. Both frameworks carry over to Physical AI, where the attacker's goal becomes physical harm rather than misclassification. Eykholt et al. [11] demonstrated physical stickers on stop signs causing autonomous vehicles to misclassify them with high confidence. Zizzo et al. [8] showed that adversarial perturbations on ICS sensor data can hide cyber-physical attacks from LSTM-based IDS, requiring on average only 2.87 of 12 sensors to be compromised to hide 23 of 24 detected attacks in the continuous regime. Anthi et al. [7] demonstrated that adversarial perturbations generated by the Jacobian-based Saliency Map Attack (JSMA) degrade Random Forest and J48 ICS IDS classification accuracy by 6–11 percentage points, with accuracy recovering under adversarial training. Pasikhani et al. [10] introduce multi-agent DRL-crafted stealthy wear-out attacks against water treatment ICS that evade contemporary AI defenses by blending in with normal operational patterns. Validated on an industry-level testbed, it is the most advanced adversarial technique against Physical AI in our corpus.

Adversarial ML attacks on Physical AI can be categorized by phase. *Training-time attacks* (data poisoning, backdoor injection) target the model before deployment. *Inference-time attacks* (adversarial examples, model inversion) target the deployed model during operation. *Cross-layer attacks* combine both: a supply-chain compromise introduces a backdoored model (L1 training-time attack) that then responds to a specific inference-time trigger (L3/L4 attack). Huma et al. [28] survey adversarial ML in IoT and ICS settings and give a detailed taxonomy.

In Physical AI the effect of an adversarial attack does not stop at the perturbed sensor; it runs through the whole kinematic chain. In deep reinforcement learning, small adversarial perturbations of an agent's observations substantially degrade the learned control policy [159]. In autonomous vehicles, LiDAR spoofing through laser replay attacks injects spoofed obstacles into the perceived point cloud [160], and the attack has worked under real-world conditions. In ICS, Stuxnet is the earliest real-world example: crafted process inputs drove a controlled system toward destructive

states while staying below the statistical detection thresholds of multiple sensors for months. Physical AI combines all three settings, so an attacker who understands the weaknesses of a learned sensor-fusion stack can produce physical effects that no single-modality defense detects.

## 5.4 AI as an Attack Tool (2025–2026)

In 2025–2026 adversaries began turning AI against Physical AI defenses. LLM-assisted penetration testing tools such as PentestGPT [12] are being adapted to industrial targets, which lets adversaries automate the search for ICS protocol weaknesses. Industrial Cyber [6] reports that the move from reactive to intelligence-driven OT security has an offensive mirror image: the AI tools used for anomaly detection are being repurposed to craft attack sequences that evade it.

Generative AI adds further options for the attacker. Large language models [108, 109] can be prompted to generate plausible attack sequences against ICS protocols [110], synthesize social engineering scripts targeting human operators, or produce deepfake audio and video commands for HRI attacks [65]. The IDSGAN framework [94] drives the detection rate of seven black-box ML-based IDS classifiers on NSL-KDD DoS traffic from roughly 73–85% to below 1.3%, an evasion increase rate above 98.5% in every case. Similar methods could be applied to OT protocols, whose statistical signatures can be learned from public ICS datasets [114, 115]. Query-efficient black-box attacks lower the query cost enough to make attacks on Physical AI controllers practical without direct model access.

With AI on both sides, we see three consequences for Physical AI security. First, attack sophistication is likely to grow faster than defender capability, because AI lowers the cost of generating new attack variants [12, 94]. Second, defenses relying on static rules or on fixed ML models trained on historical attacks will lose effectiveness over time [7, 92]. Third, L5 (knowledge stores) becomes a primary target: poisoning the invariant registry or the red-team scenario pipeline corrupts the whole feedback loop, and the system becomes more vulnerable with every update cycle.

## 5.5 LLM-Specific Attack Vectors in Physical AI

Putting LLMs inside Physical AI decision loops creates attack vectors that the ICS and robot security literature has not dealt with. We identify five LLM-specific attack classes that cut across PAIS layers.

*(1) Direct and indirect prompt injection.* White-box prompt injection attacks on embodied AI agents have been demonstrated across LLaMA, Mistral, and Vicuna models in autonomous driving, robotic manipulation, drone control, and industrial control scenarios [132]. Zhang et al. [131] demonstrate that adversarial prompts cause GPT-4o-integrated mobile robots to make dangerous navigational decisions. Perez and Ribeiro [111] established the direct form against LLMs, and Greshake et al. [149] the *indirect* form, in which attacker instructions are embedded in data the model later processes. The indirect variant is the more serious in Physical AI because it does not require access to the operator's console: an attacker who can place text in a QR-coded work order, a maintenance record, a manipulated knowledge base, or a poisoned

retrieval store can induce attacker-chosen physical actions while the system presents a plausible natural-language rationale to its human overseer. The OWASP Top 10 for LLMs [133] identifies prompt injection as LLM01:2025, the primary vulnerability class. Transfer behaves differently in embodied settings: white-box injection suffixes against embodied agents transferred well across versions within a model family but very poorly across architectures [132].

*(2) PoisonedRAG.* Inserting as few as 5 malicious documents into a corpus of millions causes an LLM-based system to return adversary-chosen answers 90% of the time for trigger queries [134]. In Physical AI contexts, a poisoned knowledge base can cause a robot to apply incorrect assembly procedures or an OT planner to recommend harmful control sequences.

*(3) Jailbreaking embodied AI.* Universal, transferable adversarial suffixes that jailbreak aligned models, including the public interfaces of ChatGPT, Bard, and Claude, have been demonstrated [135]. When such a model governs a physical system, a jailbreak can produce dangerous command sequences that no current PAIS mechanism filters: L1 attests the hardware and L2 validates physical invariants, but neither checks the *semantic safety* of an LLM-generated action sequence before it runs.

*(4) Federated model poisoning.* In federated learning for ICS intrusion detection [42, 165], a malicious participant can submit poisoned local model updates [97] that degrade the global anomaly detector while appearing to contribute honestly to the federation. The threat sits between L3 (runtime) and L5 (knowledge feedback).

*(5) Adversarial attacks on time-series IDS (JSMA).* Zizzo et al. [8] demonstrate that the Jacobian-based Saliency Map Attack (JSMA) applied to ICS sensor data hides cyber-physical attacks from LSTM-based IDS, requiring on average only 2.87 of 12 sensors to be perturbed. That budget is within practical reach, so runtime assurance built on a single detector is not enough.

Together, the five classes motivate research challenge R4b (Section 9), a research agenda for LLM-driven Physical AI that goes further than conventional adversarial ML.

# 6 Layer-Specific Systematic Literature Analysis

This section applies the PAIS taxonomy to the 312-paper corpus, one layer per subsection. Each subsection reviews the principal research streams for its layer, compares representative approaches (tabulated where studies report comparable attributes), and identifies the open problems carried into the research agenda of Section 9. Coverage differs sharply across layers: under the classification of Section 3.5, the corpus yields 85 paper–layer assignments for L1, 111 for L2, 165 for L3, 57 for L4a, and 202 for L4b, while the cross-cutting L5 is analyzed qualitatively. Performance figures quoted in this section are subject to the evidence-base limits of Section 3.6.

## 6.1 Layer 1: Trust and Execution

### 6.1.1 Hardware Roots of Trust

A *hardware root of trust* (HRoT) is a cryptographically anchored, physically tamper-resistant component at the start of the chain of trust for all software measurements. Four HRoT architectures are in production or early use for Physical AI; Table 6 compares them.

*TPM 2.0* (ISO/IEC 11889) provides Platform Configuration Registers (PCRs) for boot-time measurements, sealing, and remote attestation. Its use in OT/ICS edge nodes is growing, enabled by firmware-TPM implementations that do not require dedicated chips. SEC-TPM, built on ARM TrustZone, provides TPM 2.0 functions within a secure execution environment and integrates with VPN for secure remote connectivity. The attestation protocol produces a signed PCR quote that bundles hardware identity, firmware version, and boot-chain measurement, which is the provenance record that Layer 1 needs.

*Trusted execution environments (TEEs)* isolate AI inference workloads within CPU-level enclaves, providing confidentiality and integrity of the model and its runtime even against a compromised host OS. Menetrey et al. [13] compare Intel SGX, AMD SEV, and ARM TrustZone-A/M: SGX offers the most mature developer tooling and fine-grained process-level isolation but has been deprecated on Alder Lake and later Intel processors; TrustZone is ubiquitous on ARM edge platforms (Cortex-A and Cortex-M series) but has fragmented documentation and tooling; AMD SEV offers VM-level confidentiality suitable for cloud-hosted Physical AI inference but is not edge-native. Schneider et al.'s SoK [14] covers measurement, attestation, and sealing across the major hardware-supported TEE architectures. NVIDIA Jetson platforms offer secure firmware flashing and rollback prevention, which makes them usable as HRoT platforms for Jetson-class edge inference.

*Hardware security modules (HSMs)* protect cryptographic keys and perform signed computation in a dedicated tamper-resistant package. In Physical AI, HSMs appear in signing pipelines for model attestation (binding an inference model hash to a signed provenance record) and in OT PKI for device identity management. The throughput limitation of HSMs (typically hundreds to low thousands of operations per second) makes them unsuitable for per-inference attestation at high-frequency control loops but appropriate for model load-time signing.

*OpenTitan* is an open-source silicon root of trust designed for IoT and embedded devices [48]. Because the design is open, the attestation logic can be reviewed independently, which closed HRoT implementations do not allow. It is still early in its maturity and adoption is limited, but it shows where verifiable, open trust anchors for Physical AI are heading.

### 6.1.2 Secure Boot and Supply-Chain Integrity

Secure boot enforces a measured boot chain from Boot ROM through firmware, operating system, and AI runtime, where each stage verifies the next stage's digital signature before transferring execution. For Physical AI edge nodes, TPM PCR-based attestation can chain hardware identity to TEE identity through certificate chains that

**Table 6** Hardware attestation mechanisms: comparative evaluation for Physical AI edge deployment

| **Mechanism** | Scope | Isolation Level | Perf. Overhead | Edge Suitability | Physical AI Status | Key Limitation |
|---|---|---|---|---|---|---|
| **TPM 2.0** | Platform | Chip-level PCR | Moderate (≈5 ms per quote; see note[†]) | High (fTPM on Cortex-A) | Growing (OT edge nodes) | No AI-specific attestation standard |
| **ARM TrustZone** | CPU-level | Secure-world isolation | Very low (<0.5 ms) | Very high (all Cortex-A/M) | Ubiquitous [13] | Fragmented tooling; doc gaps |
| **Intel SGX** | Enclave (process) | Process-level enclave | Medium (5–15%) | Low (server only) | Limited in edge [13] | Deprecated on Alder Lake+; enclave-size limits |
| **AMD SEV** | VM-level | Hypervisor-mediated | Low (1–5%) | Low (server/ cloud) | Data-center AI [14] | Not edge-native; requires hypervisor |
| **HSM** | Crypto ops | Dedicated HW package | Low (async ops) | Medium (OT PKI) | OT PKI/signing | Throughput limits; not for per-inference |
| **OpenTitan** | Platform | Open-source RoT | Low [48] | High (IoT/ embedded) | Production (Google Titan, 2023+) [49] | Open-source; growing adoption in security-focused embedded systems |
| **Firmware-TPM** | Platform | SW emulation (TZ-based) | Negligible | Very high (IIoT/ embedded) | IIoT/ embedded | SW-only; weaker than discrete TPM |

[†]TPM 2.0 quote latency is *not* sub-millisecond. An end-to-end `TPM2_Quote` is reported at ≈4.8 ms in an optimized attestation stack [155]; firmware TPMs are 2.4×–15× faster than discrete parts across TPM 2.0 commands [156], so discrete TPMs are correspondingly slower; and the Linux TPM driver's polling schedule (initial 20 ms wait, doubling) adds host-observed latency [157]. Per-decision attestation is therefore not viable inside a 100 ms control budget: attestation must be amortized across decisions and cached (see Section 7).
Note: No standard mandates any HRoT mechanism for Physical AI edge inference nodes (see R8).

trace back to manufacturer roots. A remote verifier can then confirm cryptographically, before admitting any AI decision, that the edge node is running the expected firmware, OS, and model version.

Supply-chain integrity has received little attention in Physical AI. Compromising a single HSM component in a robot's control stack undermines every layer of trust above it. NIST SP 800-161 guidance on Cyber Supply Chain Risk Management (C-SCRM) is cited more and more often in Physical AI security architectures, but empirical studies of supply-chain attacks specific to robotic platforms remain scarce [71]. The distinction between *hardware* supply-chain attacks (counterfeit chips, hardware Trojans) and *software* supply-chain attacks (malicious firmware updates, compromised model weights) matters: hardware attacks are much harder to detect after deployment and cannot be fixed by a software patch.

Three supply-chain threat vectors are specific to Physical AI: (1) *Counterfeit PLC/edge-node hardware* with altered cryptographic units that generate seemingly valid but weakened attestation evidence; (2) *Malicious pre-trained model weights* distributed through model repositories, containing backdoor triggers activating under specific physical inputs; (3) *Compromised firmware update pipelines* in OT settings,

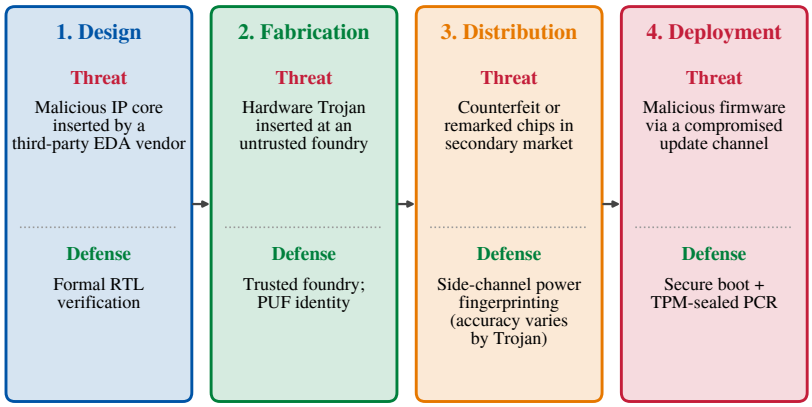



**Fig. 5** Physical AI supply-chain attack surface across four lifecycle stages. Each panel shows the characteristic threat (red label) and primary defense. Machine-learning side-channel detection is near-perfect for some Trojan benchmarks but degrades sharply for Trojans whose power signatures overlap normal operation [137].

where PLCs may accept unsigned firmware pushed through engineering workstations. None of these three vectors has a standardized mitigation in current Physical AI security frameworks.

Supply-chain risks span four stages in the Physical AI component lifecycle (Fig. 5):

(1) *Design phase*: Malicious IP core insertion by third-party EDA vendors. Formal RTL verification can detect specific trojan behaviors, but does not scale to complex SoCs.
(2) *Fabrication phase*: Malicious functionality inserted during IC manufacturing at untrusted foundries. The global concentration of advanced semiconductor fabrication (primarily Taiwan and South Korea) creates geopolitical supply-chain risk for AI accelerator chips.
(3) *Distribution phase*: Counterfeit components (relabelled chips, remarked ICs) entering the secondary market. Power side-channel analysis of received parts can reveal Trojan activity, but machine-learning detectors are reliable only when a Trojan's power signature is distinctive: on twelve AES Trojan benchmarks, accuracy ranged from 98–100% for four to roughly 50–77% for the rest [137].
(4) *Deployment phase*: Malicious firmware updates through compromised update channels. UEFI/BIOS secure boot with TPM-sealed measurements provides the primary defense; NIST SP 800-161r1 (C-SCRM) provides the supply-chain risk management framework, though neither addresses AI-model supply-chain integrity (training data provenance, model weight lineage).

### 6.1.3 Hardware Trojans

The September 2024 Lebanon pager incident, in which PETN explosives introduced through the hardware supply chain were detonated remotely, is the most extreme

physical outcome of a hardware-level attack to date [136]. Chatterjee observes that this "shows the evolving nature of cyber and hardware warfare, showing how supply chain vulnerabilities can be exploited to introduce hardware Trojans, turning devices into lethal weapons" [136].

Hardware Trojans relevant to Physical AI fall into four classes:

- *Functionality trojans*: malicious logic triggered by rare input conditions to alter actuator behavior covertly;
- *Kill switches*: covert circuitry that disables safety systems on remote command;
- *Exfiltration trojans*: covert channels transmitting model weights or control telemetry to adversaries, which enables offline model extraction;
- *Degradation trojans*: subtle performance degradation mimicking natural hardware aging, producing stealthy wear-out effects at the silicon level.

Detection methods include formal verification of RTL designs, side-channel power analysis [137], X-ray CT scanning for post-fabrication physical inspection, and AI-powered runtime monitoring [138]. None of them scales reliably to the complex SoCs used in Physical AI edge nodes. Trojan detection for AI accelerators (GPU/NPU chips) is treated as an extension of Research Challenge R2 in Section 9.

### 6.1.4 Model Integrity and Provenance

*Model attestation* extends software attestation to AI. It binds a cryptographic hash of a trained model, the model's provenance record (training data lineage, hyperparameter configuration), and its runtime measurement into a signed envelope that accompanies every inference output. The idea appears implicitly in several federated-learning security papers [15] but has not been formalized as a Physical AI standard. Bridging model attestation research and deployed TPM-based infrastructure is one of the largest open problems in the field.

Model provenance has three separate properties: *origin* (who trained the model, on what data, with what code); *integrity* (that the deployed model is the one that was trained, with no weights modified in transit or at rest); *authorization* (that the model has been reviewed and approved for the specific physical deployment context). Each property requires different cryptographic mechanisms: origin requires signed training manifests; integrity requires model hash binding in a TPM-sealed PCR [47]; authorization requires a deployment certificate issued by a validation authority.

Federated learning [95] introduces additional model provenance challenges: the trained model is an aggregate of contributions from multiple participants, and verifying that no participant injected a backdoor [97] requires cryptographic proofs of gradient integrity that remain computationally expensive in practice. Bonawitz et al. [96] demonstrate a production-scale federated learning system at Google that provides aggregation integrity through secure aggregation protocols, but their setting (mobile devices, classification tasks) is far from Physical AI OT environments (PLCs, regression and control tasks). How to adapt secure federated learning to OT remains open.

Model watermarking, which embeds imperceptible signatures in trained models so that ownership can be verified, is a complementary provenance mechanism. Unlike attestation, watermarking survives model extraction attacks [99]: an attacker who queries a watermarked model to reconstruct a surrogate may inadvertently copy the watermark, enabling the original owner to prove prior rights. In Physical AI contexts, model watermarking could be used to detect when a stolen robot controller model is deployed without authorization. This matters more as robot AI components are sold as proprietary intellectual property.

### 6.1.5 Zero Trust for Physical AI

Zero Trust Architecture (ZTA), as defined by NIST SP 800-207 [69], assumes no implicit trust even inside a network perimeter. Applied to Physical AI, ZTA means every decision request from an AI agent must present a verified identity claim (attestation) before any action is admitted. Chen et al. [16] propose ZTA extensions for 6G-connected Physical AI that integrate continuous authentication with edge inference pipelines. In a typical data flow, a model request triggers TPM quote generation, cloud attestation, and an access decision before the model is decrypted and inference proceeds.

The difficulty at the edge is that ZTA assumes low-latency connectivity to a policy enforcement point (PEP) or policy decision point (PDP). That assumption fails in air-gapped OT environments and in latency-critical control loops. Rose et al. [69] define ZTA as eliminating implicit trust in any element, node, or service inside or outside the network perimeter, requiring continuous verification. Buck et al. [70] survey practical ZTA implementations and find that the main problem in every deployment is the latency of continuous verification. The problem is worse in Physical AI, where verification latency adds directly to the control loop.

Mishra et al. [29] couple an IIoT intrusion detector to a zero-trust manager that updates device trust scores from prediction confidence, record the updates in a lightweight SHA-256 hash-chained ledger, and visualize device trust states in a digital twin; the design is centralized and evaluated in simulation on merged public datasets, so its behavior across the connectivity gaps typical of OT remains untested. Chen et al. [16] propose zero-trust architecture for 6G-connected Physical AI systems, showing that mutual TLS combined with continuous device attestation can achieve sub-10 ms policy decisions for known device identities. That is feasible for connected Physical AI systems but not for air-gapped OT. ZTA's "never trust" principle pulls against Physical AI's need to decide in milliseconds, and resolving that tension is an open systems-architecture problem. ZTA here is an *architectural pattern* for reducing implicit trust, not a guarantee that malicious decisions will be blocked: an attacker with valid credentials and a poisoned model can still issue attested decisions that pass L1 checks.

### 6.1.6 Model Extraction and Side-Channel Attacks

In addition to direct model compromise, Physical AI models face extraction attacks: systematic querying of a black-box model to reconstruct a surrogate that approximates its decision boundary [99]. In Physical AI, an extracted model can be used to (1) craft

adversarial inputs that reliably fool the target model, or (2) introduce a backdoored variant that the attacker re-submits to the target supply chain. The defense at L1 is to rate-limit and monitor inference API calls, and to require attestation of the querying entity; in other words, to apply zero trust to the model API.

Membership inference attacks [98] allow an adversary to determine whether a specific data point was used in training, leaking private operational data. Model inversion attacks [100] recover representative training samples from model confidence outputs. In Physical AI these attacks can expose proprietary operational data (process parameters, robot trajectories, patient rehabilitation data) with both commercial and safety implications. Song et al. [101] show that defenses against adversarial examples, such as adversarial training, can increase privacy leakage, so robustness and privacy pull against each other at L1.

Side-channel leakage from edge AI hardware is a distinct L1 threat. Power analysis and electromagnetic side-channel attacks can recover model weights from inference hardware [50]. Dubey et al. [51] propose MaskedNet, a masked hardware implementation that mitigates power side-channel leakage for embedded neural network inference at the cost of a 2.3× increase in area and 2.8× in latency. The threat matters most where the controller's ML model is proprietary. Leaked weights let an attacker study the model offline and craft inputs aimed at its particular decision boundaries. The resulting two-stage attack evades runtime monitoring, because the crafted input looks like legitimate traffic until the physical consequence occurs.

### 6.1.7 Publication Coverage Analysis

Fig. 6 shows publication counts in the L1 domain by sub-topic and year from the survey corpus. Hardware attestation grew the most (2020–2025), while model provenance is still sparsely studied; the field has concentrated on infrastructure-level trust rather than AI-specific trust.

## 6.2 Layer 2: Pre-Deployment Validation

### 6.2.1 Behavioral Digital Twins for Safety Validation

A *behavioral digital twin* (BDT) is a formal or semi-formal model of a physical system's expected dynamics under a given control policy, used to evaluate candidate AI decisions before physical execution. Unlike high-fidelity physics simulations, BDTs are invariant-driven: they check whether a proposed action violates a set of cyber-physical invariants (safety envelopes, energy bounds, kinematic limits, production constraints) rather than simulating exact physics. A twin that had to simulate exact physics would be too slow for pre-deployment validation and infeasible for complex multi-body systems.

Eckhart et al. [17] coined the term *Security-Enhancing Digital Twins* (SEDTs), which are designed to improve CPS security without disrupting operations. De Benedictis et al. [18] propose a Secure Cyber Digital Twin framework integrating SEDTs with anomaly detection and deploy it in cloud-managed industrial environments. More recently, Cho et al. [19] develop a Generative Digital Twin that continuously updates functional safety invariants using online learning and detects attack patterns that

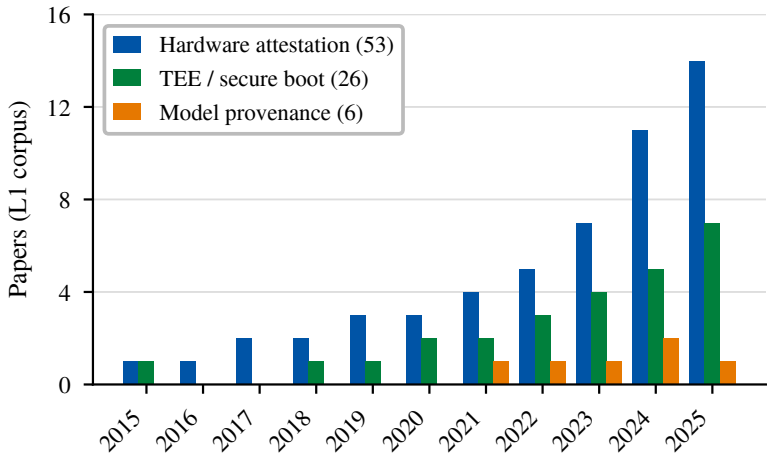



**Fig. 6** Publication trends in Layer 1 (Trust & Execution) sub-topics, 2015–2025 (survey corpus, $n = 312$). Hardware attestation grew fastest after 2020; model provenance remains sparsely studied.

**Table 7** Digital twin frameworks for Physical AI security validation

| Framework | Twin Type | Invariant Support | Online Learning | Cyber-Physical Integration | Validation Scope | Std. | Status |
|---|---|---|---|---|---|---|---|
| **Eckhart et al. SEDT (2019) [17]** | Security-Enhancing DT | Manual invariants | No | ICS/CPS | Security monitoring | No | Research |
| **De Benedictis et al. (2022) [18]** | Secure Cyber DT | Partial auto | No | CPS | Anomaly detection | No | Research |
| **Cho et al. Generative DT (2025) [19]** | Gener-ative DT | Auto-updated | Yes | CPS | Functional safety | No | Research |
| **Garadi et al (2024) [21]** | Behavioral DT | Discovered rules | Partial | ICS | Sequence anomaly | No | Research |
| **Kurup-puarachchi Trust DT (2025) [25]** | Trust DT | Trust metrics | Partial | Collab. CPS | Behavioral categ. | No | Research |
| **Xu et al (2023) [162]** | Learning DT | Curriculum rules | Yes | CPS | Rare attack detect. | No | Research |

*No BDT framework has been standardized for cross-domain Physical AI use.

were not seen when the twin was first built. Between 2018 and 2024, digital twins in cybersecurity were applied mainly to IDS, simulation testing, and legal compliance. Table 7 evaluates representative BDT frameworks for Physical AI security.

BDTs in Physical AI also need *invariant registry management*: the twin should read its invariants from a versioned, access-controlled registry instead of hard-coding them. When domain conditions change (a new product line, a new robot configuration,

**Table 8** Invariant specification languages for cyber-physical systems

| **Language / Approach** | Expressiveness | Tool Support | Temporal Props. | Spatial Props. | HRI Supp. | OT/ICS Supp. | Std. for Phys. AI |
|---|---|---|---|---|---|---|---|
| **Signal Temporal Logic (STL)** | High (continuous) | Breach [78], S-TaLiRo [79] | Yes (bounded) | Partial | Partial | Partial | No |
| **Metric Temporal Logic (MTL)** | High (quantitative time) | Uppaal [80], MightyL [104] | Yes (real-time) | Limited | Limited | Limited | No |
| **Assume-Guarantee (A-G)** | Medium (component) | AGREE [103], OCRA [104] | Partial | Yes (via geometry) | Limited | Partial | No (DO-178C only) |
| **Process Mining (event logs)** | Medium (sequences) | ProM [76], Disco | Partial (trace-level) | No | No | Yes | No |
| **Contract-based Design** | Medium (interface specs) | Simulink Design Verif. [106] | Partial | Yes | Limited | Limited | No (avionics) |

No existing language covers robotics, OT, and HRI domains simultaneously. A Physical AI Invariant Language (PAIL) addressing all three domains is identified as Research Challenge R2 (Section 9); it is a design target, not a deployed tool.

a new regulatory requirement), the validation logic can then be updated without a software release.

The broader digital twin literature [56] provides the engineering foundation for BDT development. Grieves originally proposed the digital twin as a virtual replica enabling life-cycle management; Fuller et al. [56] identify enabling technologies (IoT, AI, cloud computing) and open challenges, including security and privacy. From a security standpoint, the Eckhart-Ekelhart SEDT (Security-Enhancing Digital Twin) [17, 57] is the closest prior work: it uses a digital twin as a runtime security monitor that maintains a virtual state of the physical system and detects divergence from expected behavior.

Digital-twin-based security is exposed to *twin poisoning*: if the twin's state model or invariants can be corrupted by an adversary, the twin will validate harmful AI decisions rather than detecting them. Van der Horn and Mahadevan discuss digital twin fidelity and identify state estimation errors as the primary source of false negatives in monitoring applications. Mitigating twin poisoning means applying L1's attestation and access-control mechanisms to the twin infrastructure itself, so securing a BDT is a smaller copy of the PAIS problem inside the validation layer.

### 6.2.2 Invariant Specification Languages

There is no standard language for expressing cyber-physical invariants. Current practice mixes several formalisms, and none of them covers robotics, OT, and HRI together. Table 8 compares the main approaches.

Signal Temporal Logic (STL) expresses time-bounded properties over continuous signals and is widely used in autonomous vehicle verification. Metric Temporal Logic

**Table 9** Formal verification tools for neural network controllers: applicability to Physical AI

| Tool | Approach | Max Network | Property Types | Latency | Physical AI Application | Key Limitation |
|---|---|---|---|---|---|---|
| **Reluplex [23]** | SMT-based | $\sim 10^3$ params | Safety, reachability | Hours–days | Small NNs only | Exponential scaling; ReLU only |
| **AI$^2$ [24]** | Abstract interp. | $\sim 10^4$ params | Robustness, bounds | Minutes | Certification contexts | Over-approximation; false negatives |
| **NNV [22]** | Reachability | $10^5$–$10^6$ (exact, tight bounds) | CPS safety specs | Minutes–hours | Learning-enabled CPS | Exact bounds only for moderate networks |
| **CROWN [52]** | Bound propagation | $10^6$–$10^8$ (loose bounds) | Lipschitz bounds | Seconds | Robustness certification | Bounds loosen at scale; safety props limited |
| **Marabou [53]** | SMT+simplex | $\sim 10^4$ (exact); $10^5$ (incomplete) | Piecewise-linear NNs | Hours–days | Avionics, robotics | Exponential worst-case; ReLU only |
| **Runtime Verif. (monitors)** | Monitor-based | Unlimited | Temporal specs | $<$1 ms | Embedded CPS | Detection only; no offline proof |
| **Deployed PhysAI model [GAP]** | N/A | $10^6$–$10^9$ params | Tight safety props required | $<$50 ms required | Target system | **No tool provides tight guarantees at this scale** |

*Scale* column denotes network sizes at which the tool provides *tight, useful* safety certificates, not the raw parameter ceiling at which it can run. CROWN runs on $10^8$-param networks but its bounds become too loose to certify Physical AI safety margins. The gap is in *certification tightness*, not parameter count as such.

(MTL) adds real-time quantitative bounds useful for robotic motion planning verification. Assume-Guarantee (A-G) contracts specify each component's assumptions about its environment and guarantees about its outputs, used in safety-critical avionics (DO-178C). Process mining approaches convert OT device logs to event logs to discover behavioral models that surface sequence-level anomalies [20, 21]. Contract-based design tools such as Simulink Design Verifier support OT/robotics workflows but are not standardized across domains.

We propose a Physical AI Invariant Language (PAIL) as a design target for Research Challenge R2. PAIL would need to support: (1) real-time temporal properties (MTL/STL-family) for control-loop invariants; (2) spatial constraints for safety envelopes and proximity zones; (3) semantic properties representing production intent; (4) human-authority constraints for HRI contexts; and (5) evaluability by both digital twins (pre-deployment) and runtime monitors (real-time).

### 6.2.3 Formal Verification of AI Controllers

Formal verification of neural network controllers uses reachability analysis, satisfiability modulo theories (SMT), abstract interpretation, and runtime verification. Table 9 compares the main tools and how far each applies to Physical AI, and Fig. 7 shows the gap between what the tools can certify and the model sizes actually deployed.

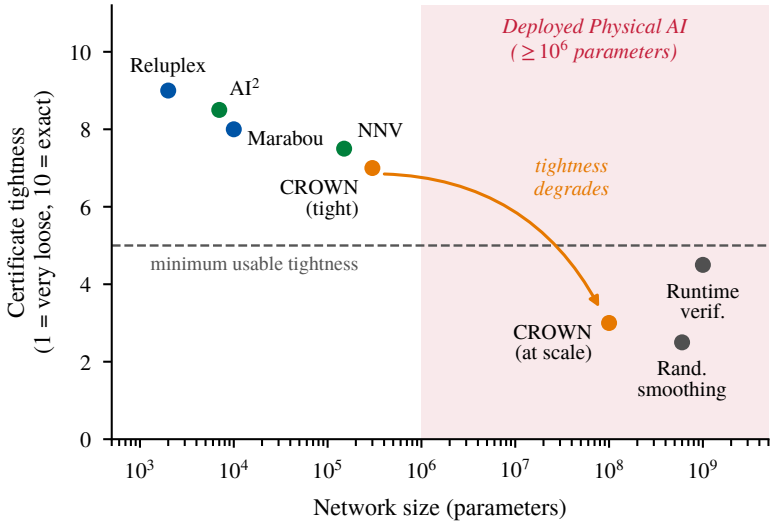



**Fig. 7** Formal verification tightness gap. The limiting factor is certificate tightness, not raw parameter count: CROWN processes $\geq 10^8$-param networks but its bounds degrade below usable certification quality at deployed Physical AI scale ($\geq 10^6$ parameters). The shaded region marks the unsolved space; the dashed line marks the minimum usable tightness.

*Reachability analysis* computes the set of states reachable by a neural network controller from a given initial set; NNV [22, 105] provides tight exact reachability for moderate-size networks; CROWN [52] provides bound propagation that scales to large networks but at the cost of increasingly loose over-approximations. *SMT-based verification* encodes neural network properties as satisfiability modulo theory constraints; Reluplex [23, 54] was the first such approach, and Marabou [53] extended it with floating-point support. Both are exact, and both have exponential worst-case complexity. *Abstract interpretation* provides sound over-approximation; AI$^2$ [24] achieves the best scalability-tightness tradeoffs in VNN-COMP competitions [55]. *Runtime verification* [104, 106] monitors execution against formal specifications and sits between Layers 2 and 3.

The scalability problem is *not* that tools cannot run on large networks. Bound-propagation methods of the CROWN family scale to networks with millions of parameters, although this was established by later work in that lineage rather than by the original CROWN paper [52], whose evaluation used small MLP and CNN benchmarks; the GPU-accelerated branch-and-bound descendants [158] are the correct attribution for large-scale demonstrations. At the scales that matter for Physical AI, current tools either give bounds too loose to certify the narrow safety margins required (relaxation-based methods) or become computationally intractable (exact SMT methods). This *tightness gap*, rather than a parameter ceiling, is the central open problem of the layer (R1). Compositional verification [103] and mixed approaches, which verify safety-critical components exactly and certify larger components statistically, look like the most practical routes forward.

### 6.2.4 Pre-deployment Red Teaming

Red teaming (systematic adversarial testing before deployment) is well established in IT security but still new to Physical AI. Three difficulties stand out.

- *Physical scenario generation*: red-team scenarios must cover physical failure modes (hardware faults, environmental perturbations) in addition to cyber attacks. Simulation-based testing [75] (DeepTest for autonomous vehicles) shows that learned test generation finds coverage gaps in neural controller behavior that manual scenario design misses.
- *Adversarial input generation for physical sensors*: generating realistic physical adversarial examples (LIDAR point-cloud manipulation, GPS spoofing trajectories) requires access to physical or high-fidelity simulation testbeds. Evtimov et al. [112] demonstrate that adversarial patches remain effective across scale, viewpoint, and lighting variation, which is a good reason to red-team robot vision systems physically.
- *LLM-assisted red teaming*: PentestGPT [12] and similar tools are beginning to be applied to OT environments, which helps defenders and attackers alike. In controlled experiments, Happe and Cito [110] found LLM-driven penetration testing comparable to junior human testers.

Linking red teaming to runtime evidence closes a loop: every confirmed attack observed at runtime becomes a new pre-deployment test case. The DRL-based attack generator of Pasikhani et al. [10] could automate this scenario generation. It learns minimal perturbations that evade the current detection configuration, which are the test cases pre-deployment validation most needs.

## 6.3 Layer 3: Runtime Assurance

### 6.3.1 Anomaly Detection in Physical AI

Runtime anomaly detection compares observed telemetry against an expected-behavior profile derived from the admitted action and validation report. We group the approaches into five paradigms; Table 14 gives an extended comparison.

*(1) Statistical process control (SPC).* CUSUM and EWMA charts have been applied to ICS process variables since the 2000s. They provide bounded latency ($<$1 ms) but limited sensitivity to structured attacks that stay within statistical bounds.

*(2) Machine learning.* Supervised (Random Forest, SVM), unsupervised (Autoencoder, Isolation Forest), and semi-supervised (One-Class SVM) models applied to network traffic and sensor telemetry. Anthi et al. [7] show these are vulnerable to adversarial evasion in ICS settings. Randomized CNN variants have reported better results on SCADA datasets.

*(3) Deep learning.* LSTM and GRU-based sequence models detect temporal anomalies in control-loop telemetry [8]. Jia et al.'s survey of deep anomaly detection [26] covers the time-series methods in detail. Federated anomaly detection [15] allows privacy-preserving learning across sites when raw OT data cannot leave the plant.

**Table 10** Edge anomaly detection paradigms on the ICS benchmarks SWaT, BATADAL, and HAI. All $F1$ ranges come from evaluations against static, manually scripted cyber-physical anomalies. The benchmarks contain no adaptive, AI-generated evasion, so the ranges are upper bounds under idealized threat conditions.

| Method | F1 (%) | Latency | Memory | Adv. robustness |
|---|---|---|---|---|
| CUSUM/EWMA | 72–81 | $< 1$ ms | Minimal | Low |
| Random Forest | 88–94 | 5–20 ms | 50–200 MB | Medium |
| LSTM/GRU | 90–97 | 20–80 ms | 100–500 MB | Low |
| Autoencoder | 85–93 | 10–40 ms | 50–300 MB | Low |
| DT-based | 91–96 | 30–100 ms | Varies | Medium |
| Invariant rules | 95–99[†] | $< 1$ ms | Minimal | High |

[†]For constraint-violating attacks only; ineffective against novel attacks.
F1 ranges are min–max across studies in the corpus reporting each method on the named datasets; they are not confidence intervals. Range width reflects cross-study variation, not within-study variance.

*(4) Digital twin-based monitoring.* The twin runs in parallel with the physical system; deviations between twin prediction and physical observation trigger alerts. Kuruppuarachchi et al. [25] introduce trust-based DT behavioral categorization for collaborative ecosystems. Xu et al. [162] use curriculum learning for DT-based anomaly detection and improve detection of rare attack patterns.

*(5) Rule-based invariant monitoring.* Hard constraints (process variable bounds, command rate limits, safety envelope violations) are checked in real time. These are not ML-based but are complementary to learned approaches and are codified in IEC 62443-3-3 security requirements.

### 6.3.2 Edge-Constrained Anomaly Detection

Physical AI runtime assurance has to run at the edge, close to the actuator, with bounded latency (typically <50 ms for fast control loops) and without relying on cloud connectivity. Accuracy suffers as a result: complex ML models detect more but need more compute and add latency. Table 10 compares representative edge anomaly detection approaches.

The table hides a pattern that recurs across the runtime assurance literature. Deep sequence models such as LSTMs score well under baseline evaluation and then degrade sharply against an adaptive attacker. A network that easily flags fuzzer traces or crude sensor spoofing can be pushed into confident misclassification by a targeted Jacobian-based Saliency Map Attack (JSMA) that perturbs fewer than 25% of the active sensor tags [8]. The figures in Table 10 should therefore be read as upper bounds under idealized, non-adversarial conditions, not as evidence of resilience against current offensive AI techniques.

The underlying problem is a three-way trade-off between accuracy, latency, and resources. LSTM/GRU models reach the highest detection accuracy (90–97%) but need 20–80 ms and 100–500 MB of memory, which rules them out on Cortex-M class hardware. Invariant rules detect known attack classes almost perfectly in under 1 ms but miss anything outside their rule set. Nobody has yet shown accuracy above 90% at under 50 ms and under 10 MB on embedded hardware, which is what Physical AI at the edge requires (Research Challenge R3).

On resource-constrained Cortex-M microcontrollers (memory budgets frequently under 256 KB RAM), TinyML techniques, including quantization, pruning, and 1D Convolutional Neural Networks or lightweight autoencoders, enable ML anomaly detection directly on the sensor node [66, 67]. Efficiency does not bring robustness, though: JSMA applied to ICS sensor data can force Random Forests and LSTMs to misclassify malicious behavior as benign with high confidence by perturbing fewer than three sensors [8].

*STL hardware monitors* are one of the more promising ways out of this trade-off. FPGA implementations of STL monitors evaluate in nanoseconds and are as hard to evade as hard invariant rules, because they encode domain knowledge that statistical evasion cannot invalidate [144]. Predictive STL monitoring uses model predictions of future system states to warn before a violation occurs, which allows containment to start early [145]. Robust STL monitors account for sensor noise and calibration error instead of assuming that sensor readings equal ground truth [146]; in Physical AI, where sensors do degrade in service, this matters.

*Federated anomaly detection* across distributed Physical AI deployments works around the privacy constraint that prevents raw OT telemetry from leaving individual facilities. DeepFed trains a CNN–GRU intrusion detector across multiple industrial CPS owners and protects the exchanged model parameters with a Paillier-based secure communication protocol [42]; Ruzafa-Alcázar et al. evaluate privacy-preserving federated intrusion detection for the industrial IoT [165]; and Mothukuri et al. apply federated anomaly detection to IoT attacks [15]. Keeping raw data on-site is data localization, not a formal privacy guarantee; a single malicious participant can poison the shared model [97], and, as with the detectors of Table 10, reported results come from benchmarks whose attacks were not generated adversarially.

### 6.3.3 Adversarial Evasion of ML-Based IDS

ML-based runtime anomaly detectors can be evaded. Anthi et al. [7] demonstrate that JSMA-generated adversarial perturbations degrade Random Forest and J48 ICS IDS classification accuracy by 6–11 percentage points; adversarial training restores performance, at a cost in clean-data detection. Zizzo et al. [8] show that only 2.87 of 12 sensor readings need to be compromised to hide 23 of 24 detected attacks from an LSTM-based IDS in the continuous regime; "full" evasion is not claimed by the original study and is not claimed here. Pasikhani et al. [10] demonstrate DRL-crafted sequences that evade autoencoder-based detectors by mimicking normal operational variance.

The theory of adversarially robust anomaly detection is less developed than that of robust classification. For classifiers, Madry et al.'s PGD training [46] provides a principled, if computationally expensive, approach to robust training. For anomaly detectors, the absence of labeled attack data during training means robust training cannot be performed in the standard way. Mirsky et al. [92] propose Kitsune, an online autoencoder ensemble trained on benign traffic; subsequent adversarial-ML work, not the original paper, established that knowledge of the ensemble architecture permits construction of adversarial traffic that eludes all of its members [93, 94]. Lin et al. [94] generalize this to arbitrary ML-based IDS using GANs to generate evasive traffic

without knowledge of the detector's internals. Qiu et al. [93] survey adversarial attacks against network IDS in IoT systems, finding that evasion success rates above 80% are routinely achieved with fewer than 100 model queries.

Three things follow. First, runtime assurance cannot rest on one detection paradigm. It needs hybrids that pair invariant rules (hard to evade, limited coverage) with ML detectors (broad coverage, easy to evade). Second, work on adversarially robust anomaly detection has to use the Physical AI threat model. Physics-based invariant rules [31] help here, because an attacker cannot easily invalidate them without also violating physical laws. Third, the detector itself needs L1 attestation, or an attacker can simply poison it.

### 6.3.4 Containment and Response

When a runtime deviation crosses a threshold, runtime assurance has to take a *local containment action* without waiting for cloud coordination. We define a four-level response hierarchy aligned with ISO 10218:2025 and IEC 62443-3-3 SR 7.1:

- *L3.R1, alert*: log the deviation; continue under enhanced monitoring; send operator notification. Triggered when STL robustness metric is negative but magnitude is below the safety-critical threshold.
- *L3.R2, slow stop*: reduce actuator velocity to zero over a validated deceleration profile (ISO 10218 Category 1 stop). Triggered when robustness magnitude exceeds the first threshold.
- *L3.R3, safe state*: move to predefined safe configuration (robot home position, valve closed, conveyor stopped); quarantine affected AI module pending re-attestation at Layer 1.
- *L3.R4, isolation*: remove node from control network; escalate to human operator; activate emergency fallback controller. Triggered for the most severe deviations or repeated L3.R2 events.

The choice between R1–R4 is severity-graded using the same STL robustness metric that detected the deviation: negative robustness magnitude maps to response level through a calibrated threshold table (Fig. 8). The response must itself be validated: a containment action that causes a secondary hazard (a sudden stop that overloads a mechanical component, for example) is a vulnerability in its own right. Ames et al. [102] propose control barrier functions (CBFs) as a formal mechanism for computing provably safe containment trajectories.

## 6.4 Layer 4a: Human–Robot Interaction Security

### 6.4.1 The HRI Attack Surface

Human–robot collaboration adds a social attack surface that classical IT and OT security do not have. Haskard and Herath [4] give the most thorough treatment so far of how safety, trust, and cybersecurity interact in cyber-physical systems. Table 11 and Fig. 10 classify the HRI-specific attacks, and Fig. 9 places the four main attack quadrants around the human–robot workspace.

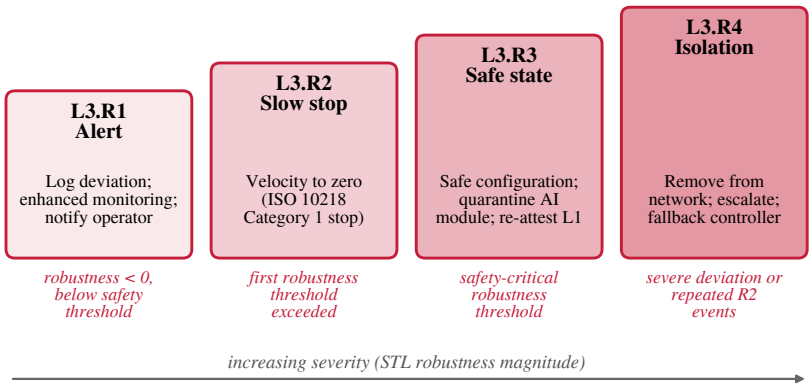



**Fig. 8** Layer 3 containment response hierarchy (L3.R1–R4). Response level is determined by the STL robustness magnitude of the detected deviation. Increasing box height and opacity reflect increasing severity. L3.R4 (Isolation) triggers on severe or repeated L3.R2 events.

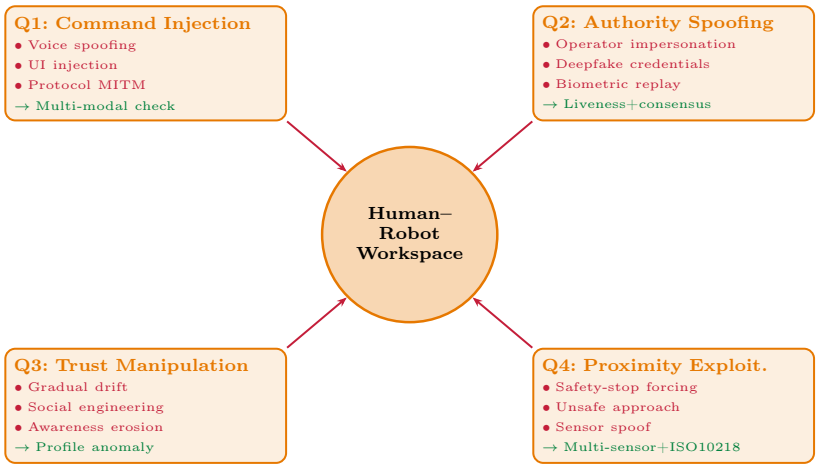



**Fig. 9** HRI attack surface: four attack quadrants around the central human–robot workspace. Solid arrows (→) indicate attack vectors (• items); dashed → prefix in each box indicates the defense approach. Color coding follows the PAIS palette; attack items are additionally distinguished by bullet symbol for greyscale legibility.

### 6.4.2 Trust Manipulation: The Hardest HRI Attack

Trust manipulation is the hardest HRI attack to detect because it acts on the operator's mental model of the robot rather than on any digital channel. The adversary changes the robot's behavior gradually, with every change within policy, until the operator's situational awareness has eroded. Once the operator no longer sees the robot as a possible threat, the adversary issues the command that causes harm.

The attack is analogous to social engineering of humans in cyber security, but the target's "vulnerability" is the inherent human tendency to build mental models of autonomous agents as intentional actors and to extend trust once initial trust is

**Table 11** Proposed HRI-specific attack taxonomy (pending empirical validation): attack classes, detection difficulty, and defense coverage. Column "Evidence" classifies each attack class as: D = Demonstrated in empirical study; T = Theorized; A = Analogous evidence from adjacent domain. Classes marked T or A require empirical study before they can be treated as established attack vectors (Research Challenge R4).

| **Attack Class** | Attack Technique | Primary Target | Physical Consequence | Det. Difficulty | Existing Defense | Coverage | Ev. |
|---|---|---|---|---|---|---|---|
| **Command Injection** | Voice spoofing, UI injection, protocol MITM | HRI command channel | Unsafe robot action | Medium | Multi-modal verification | Moderate | A |
| **Authority Spoofing** | Operator impersonation, deepfake credentials, biometric replay | Authorization system | Privilege escalation | High | Liveness detection, consensus | Low | A |
| **Trust Manipulation** | Gradual behavior drift, social engineering, situational awareness erosion | Human operator cognition | Future unsafe compliance | Very High | Profile anomaly detection | Very Low | T |
| **Proximity Exploitation** | Safety-stop forcing, unsafe approach, proximity sensor spoof | ISO 10218 safety envelope | Production disruption / collision | Medium | Multi-sensor fusion | Low | A |
| **Deepfake Commands** | AI-generated synthetic voice / gesture / video | HRI input channel | Safety-critical misdirection | Very High | Deepfake detection, biometric fusion | Very Low | A |
| **Coerced Operator** | Threatened / coerced human issuing valid commands | Human operator (physical) | Any physical action | Very high | Dual-operator consensus, behavioral baseline | None | T |

Det. = detection. Evidence: D=demonstrated on a Physical AI platform; A=analogous evidence from an adjacent ICS, IT, or non-AI robotic system; T=theorized, no public demonstration.
Defense coverage "None" means *proposed but unevaluated*, not absent from the literature: for the coerced-operator row, dual-operator consensus and behavioral baselining are described in the literature but no published evaluation measures their effectiveness against a coerced but legitimately authenticated operator.
Authority spoofing, trust manipulation, and deepfake commands are studied far less than their operational severity warrants (cf. R4).

established. A meta-analysis of 29 studies by Hancock et al. [59] found that robot performance is the primary determinant of operator trust, which means an attacker can sustain high operator trust while gradually escalating harmful behavior as long as the behavioral changes remain within the operator's performance-expectation bandwidth. Research on emergency evacuation by Robinette et al. [58] further shows that under high-stress conditions humans comply with robot guidance even when the robot has previously behaved erratically. Under time pressure, overtrust runs deep, and that gives the attacker a window.

Defenses against trust manipulation must operate at the *sequence level*: individual commands are valid, so single-command validation catches nothing.

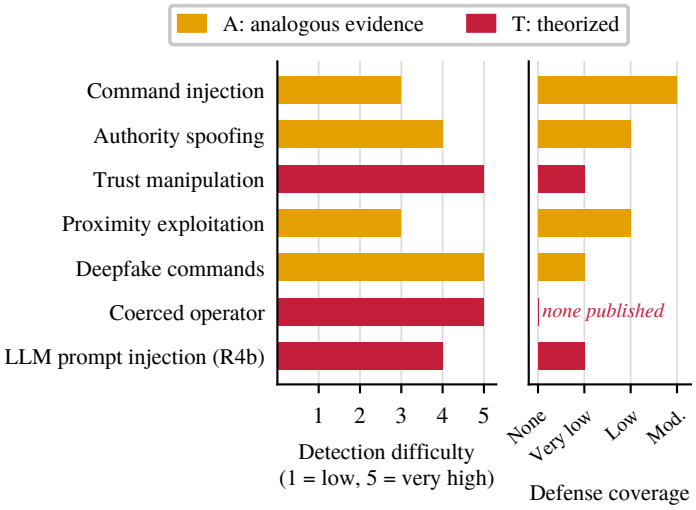



**Fig. 10** HRI attack taxonomy (seven classes, including the R4b LLM prompt-injection class). Left: detection difficulty (1 = low, 5 = very high). Right: defense coverage on the scale of Table 11 (None, Very low, Low, Mod. = moderate). Bar color gives the evidence grade (orange = analogous, red = theorized); no class is graded as demonstrated on a Physical AI platform.

### 6.4.3 Deepfake and Synthetic Media Threats to HRI

Zhao et al.'s survey of AI security in CPS [27] names deepfakes (AI-generated synthetic audio, video, or gestural commands) as a growing threat to CPS AI agents. In HRI, a deepfake voice command accepted by a compromised voice recognition system could make a cobot act unsafely. The survey argues that the interfaces between AI agents and their environment are where synthetic content can deceive machines and humans at the same time.

The deepfake generation literature [64, 65] has documented rapid capability increases driven by GANs and diffusion models. Face swapping, voice cloning, and full-body motion synthesis are now achievable in near-real time with consumer-grade hardware, which lowers the bar for HRI deepfake attacks. Tolosana et al. [65] survey detection approaches: passive artefact detection is accurate on known generation methods but fails on new generators. For HRI security, this means detectors have to be retrained adversarially as generation technology moves on.

Deepfakes are harder to detect than conventional command injection because the injected content is semantically valid (correctly formatted, acoustically consistent with the operator's voice, temporally coherent) and only distinguishable from genuine commands through biometric liveness detection or cross-modal inconsistency. An adversarial generator can be optimized specifically against the liveness detector, and in that contest a defender relying on one modality is unlikely to win.

Multi-modal authentication, which requires voice, gesture, gaze, and physiological signals to agree, is the strongest defense proposed so far. The adversary then has to synthesize consistent behavior on every channel at once, which is computationally

demanding and gets easier to detect as modalities are added. Nonverbal cues such as gesture, gaze, and timing carry communicative meaning in HRI [63], so the consistency of those cues with speech is a candidate lightweight check against synthetic commands on resource-constrained edge platforms; its effectiveness as a deepfake detector in HRI settings has not yet been evaluated.

### 6.4.4 Intent Validation Approaches

Validating human intent in HRI requires distinguishing *genuine, uncoerced commands* from spoofed, coerced, or erroneous ones. Four approaches have been proposed:

- *Multi-modal intent verification*: requiring congruence across voice, gesture, gaze, and posture before executing safety-critical commands [61].
- *Physiological liveness detection*: checking that biometric signals (heart rate variability, skin conductance) are consistent with voluntary, uncoerced operation.
- *Semantic anomaly detection on HRI sequences*: flagging command sequences that are atypical for a given operator's historical profile.
- *Consensus protocols*: requiring dual-operator confirmation for high-consequence actions (analogous to two-person integrity in nuclear settings).

Human factors research explains why intent validation is necessary and also why it is hard. Hancock et al.'s finding that trust tracks robot performance more than interaction quality or anthropomorphism [59] is what makes the gradual escalation described above possible. Short et al. [60] found that a robot that cheats at a simple game elicits greater social engagement and stronger attributions of mental state than one that plays fairly: people readily read robot behavior as intentional, which is the attribution a trust-manipulation attack exploits. Nikolaidis et al. [61] show that human-robot mutual adaptation in collaborative tasks produces convergent behavioral patterns, which an adversary with access to interaction logs could model and exploit.

### 6.4.5 Cobot Safety Standards as a Security Baseline

International cobot safety standards are not security standards, but they can limit the physical consequences an attack can reach. The limit is real. It is also narrower and more conditional than security arguments built on it usually admit, and stating it loosely overstates the protection by a wide margin.

The relevant limits are the biomechanical limits of ISO/TS 15066 Annex A, which specify maximum permissible contact force and pressure for 29 body locations grouped into 12 body regions, derived from pain-onset research [85]. They are *body-region specific*, not a single ceiling: quasi-static (clamping) maxima range from 65 N at the face and 130 N at the skull and forehead, through 140 N at the chest and at the hands and fingers, to 220 N at the thighs and knees, and transient (free-impact) limits are set at approximately twice the corresponding quasi-static values [85]. In the 2025 revision this content was carried into ISO 10218-2:2025 as informative Annex M, referenced normatively from the main text [154].

Three scoping conditions bound any security argument built on these limits. First, they apply only to *power-and-force-limiting* (PFL) collaborative applications, not to

**Table 12** HRI safety standards mapped to Physical AI security functions

| Standard | Year | Security-relevant provision |
|---|---|---|
| ISO 10218-1/2 | 2025 | Functional safety (with robot classification), safety-related cybersecurity, collaborative applications (absorbs ISO/TS 15066) |
| ISO/TS 15066 | 2016 | Annex A body-region PFL force/pressure limits (quasi-static 65–220 N by region); SSM separation distance. Content carried into ISO 10218-2:2025 Annex M |
| ISO 13849 | 2023 | Safety function PLd/e; control reliability |
| IEC 62061 | 2021 | SIL for electronic safety-related control systems |
| ANSI/A3 R15.06 | 2025 | US adoption of ISO 10218-1/2:2025; supersedes ANSI/RIA R15.06-2012 |
| IEC 62443-4-2 | 2019 | Component security for industrial automation |
| NIST SP 800-82r3 | 2023 | OT security guide; applies to robot controllers |

ISO 10218-compliant robot installations in general: a robot operating under safeguarded stop, hand guiding, or speed-and-separation monitoring is not force-limited at all, and in the 2025 revision the collaborative property attaches to the *application*, not to the robot. Second, compliance is a property of the integrated cell, verified by measurement at the contact point identified by the risk assessment, and is not inherited from a robot's certification. Third, and decisively under an adversarial threat model, the limits are enforced by safety functions that are themselves inside the attack surface: an adversary who compromises the safety-related control system, spoofs the proximity sensing that selects the operating mode, or induces a transition out of PFL mode is no longer bounded by them.

The defensible formulation is therefore conditional. *Where* a collaborative application is operating in PFL mode *and* its safety-related control system retains integrity, the Annex A limits bound the injury severity reachable by a malicious command. Neither conjunct may be assumed under an adversarial threat model, and the second is precisely what a capable attacker targets. Safety-envelope compliance is consequently a bound on *accidental* contact severity and not a security control; treating it as one is the specific error this section is intended to correct. Table 12 maps the key standards to their security-relevant provisions.

The 2025 revision of ISO 10218 narrows the gap between safety standards and security requirements without closing it. ISO 10218-1:2025 and ISO 10218-2:2025 were published in early 2025, superseding the 2011 editions. The revision clarified functional safety requirements, introduced a robot classification scheme with matching functional-safety obligations, absorbed the collaborative-application content of ISO/TS 15066, and (most relevant here) added requirements for cybersecurity *to the extent that it applies to industrial robot safety* [153, 154]. Unlike the 2011 editions, the 2025 series therefore does contain cybersecurity requirements.

What the 2025 series still does not supply is an *adversary model*. Its cybersecurity clauses are safety-motivated: security appears as a means of preserving the integrity of safety functions, scoped to robot safety rather than as a general security framework. The series does not specify a threat actor, attacker capability assumptions, an attack-surface enumeration, or verification obligations against an active adversary. Three consequences follow for Physical AI. The standard does not address the case

where a security compromise causes a safety stop to fail, or to trigger inappropriately as a denial-of-service vector (proximity exploitation, Table 11). It does not address attacks on the cognitive channel between operator and robot, which bypass the safety-related control system entirely rather than defeating it. And it does not address the semantic safety of machine-generated action sequences, the L4a attack class examined in Section 6.4. The residual gap is therefore not "no cybersecurity requirements" but "safety-motivated cybersecurity requirements without an adversary-driven threat model".

### 6.4.6 LLM-Mediated Robot Commands and Prompt Injection

Since 2022, Physical AI systems have increasingly used large language models as natural-language instruction interfaces. RT-2 [150] grounds language and vision inputs directly to robot actions; PaLM-E [151], SayCan [152], and similar systems translate natural-language task descriptions into robot motion plans. These LLM-based interfaces open a new kind of L4a attack surface: *adversarial prompt injection*, in which crafted text overrides the instruction interface's intended behavior. The attack mechanics, in both their direct and indirect forms, are treated in Section 5 (LLM attack class 1) and are not repeated here; this subsection addresses what is specific to the human–robot interaction layer.

The HRI-specific problem is that prompt injection defeats the layer's authentication model without attacking it. L4a intent verification, as currently conceived in the literature, establishes *who* issued a command through biometric or credential-based means. Prompt injection leaves that question correctly answered: the command genuinely originates from the authenticated planning pipeline, and the operator who authorizes it is the legitimate one. What has been substituted is the *semantics* of the instruction, and no PAIS layer as currently constituted validates that. L1 attests the robot's hardware and firmware, not the meaning of its instructions. L2 validates against physical invariants, but an LLM-controlled robot can select an attacker-chosen trajectory that satisfies every invariant while performing an unintended task. L3 observes execution against an expected behavioral profile that was itself derived from the substituted instruction.

The resulting defense requirements are correspondingly L4a-specific: instruction-integrity checking that binds a plan to an authenticated *intent* rather than to an authenticated *sender*; sandboxed evaluation of model-generated action sequences before physical execution; and dual-approval for sequences outside the operator's established command history. This class is Research Challenge R4b in Section 9, requiring an empirical threat model and defense evaluation distinct from the biometric-centered R4 attacks.

## 6.5 Layer 4b: Operational Technology and ICS Security

### 6.5.1 Threats Arising from IT/OT Convergence

Operational technology (OT) is the hardware and deterministic control software that runs physical processes in manufacturing, energy generation, water treatment, and

transport logistics. It was long protected by physical air gaps and proprietary protocols. Industry 4.0 has connected these systems to enterprise IT networks, cloud optimization services, and industrial IoT (IIoT) platforms, and the old boundaries have gone without a matching improvement in security engineering. Deployed industrial control systems (ICS) were designed to run deterministic routines in isolated networks; defending against an active adversary was not among their design goals.

The consequences of this convergence are measured partly by incident-response telemetry and partly by vendor threat reporting. The two carry different evidentiary weight, so we keep them apart.

*Incident-response telemetry.* Dragos recorded 1,693 ransomware attacks on industrial organizations in 2024, an 87% year-on-year increase, and reports that of the ransomware incidents it responded to, 75% produced a partial and 25% a full shutdown of operational technology [119].

*Vendor-reported estimates.* The following figures derive from commercial threat-intelligence publications that do not disclose sampling frames or denominators, and we report them as indicative rather than measured:

- a reported 49% expansion in state-aligned offensive campaigns against energy, transport, and water utility networks [6];
- a reported estimate that roughly three quarters of physical plant disruptions originate as enterprise IT perimeter breaches that move laterally across the IT/OT boundary [161];
- a reported observation that process-data manipulation is now the most prevalent adversary technique in manufacturing, transit, and energy utilities [161].

Grey literature is retained in this survey where it is the only available source of operational incident data, a common condition in OT security, but it is weighted below peer-reviewed and incident-response sources throughout: no gap finding, layer count, or research-challenge prioritization in this survey rests on a vendor-reported statistic.

Attackers also have OT-aware toolkits: *PIPEDREAM* (also known as *INCONTROLLER*), built to manipulate Schneider Electric and OMRON PLCs; *Industroyer2*, aimed at IEC 104 power distribution substations; and *CosmicEnergy*, which uses weaknesses in legacy Modbus to toggle power line switches [161]. Toolkits like these lower the skill needed to cause physical effects, so well-resourced criminal groups can now attempt operations that used to be within reach of state military units only.

### 6.5.2 Semantic Integrity: Validating Command Sequences

The most sophisticated documented cyber-physical campaigns, among them Stuxnet, Triton/TRISIS, and the original Industroyer grid attack, have one feature in common: each protocol command the adversary injected stayed within acceptable mechanical parameter bounds. The damage came from the *sequence and composition* of those commands over time. We call this the *semantic integrity problem*. A checker that examines commands one at a time cannot detect it. Fig. 11 illustrates the problem with an adversarial wear-out attack.

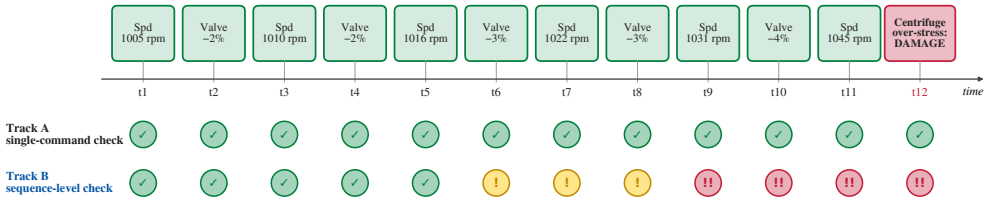



**Fig. 11** OT semantic integrity attack, modeled on Stuxnet-style wear-out. Each command is individually valid (Track A: all green ticks). Only a sequence-level semantic checker (Track B) sees the cumulative pattern, with an amber warning at t6 and a red alert at t9, before the centrifuge is damaged at t12. Single-command validators cannot catch this class of attack.

Formally, let $\mathcal{C} = \{c_1, c_2, \ldots\}$ be the set of atomic ICS protocol commands, and let $\mathcal{V}(c_i)$ be a Boolean single-command validity predicate that checks format and local parameter bounds. A command sequence $\sigma = c_{t_1}, c_{t_2}, \ldots, c_{t_n}$ is *individually valid* if and only if $\forall i,\ \mathcal{V}(c_{t_i}) = \top$. The *semantic integrity problem* is to construct a trace monitor $\mathcal{M}(\sigma)$ that detects sequences $\sigma$ satisfying

$$(\forall i,\ \mathcal{V}(c_{t_i}) = \top) \wedge \phi(\sigma) = \bot \tag{1}$$

where $\phi(\sigma)$ is a Signal Temporal Logic (STL) specification [107] of the production intent. In a wear-out scenario, $\phi$ could be $\square_{[0,T]}(\mathit{speed} \leq \omega_{\max} \wedge \mathit{cumulative_stress} \leq S_{\max})$. A bounds checker misses the violation because every command $c_{t_i}$ keeps instantaneous speed below $\omega_{\max}$ while cumulative stress climbs past $S_{\max}$ over the sequence. Research Challenge R5 starts from this definition.

The stealthy wear-out attacks of Pasikhani et al. [10] are an instance of exactly this problem. Their deep reinforcement learning (DRL) agents generate command streams that accelerate component wear and degrade product quality while staying within normal operating limits, and on an industrial water treatment testbed these streams evaded current AI-based anomaly detectors. Zizzo et al. [8] show a related result for detection: white-box adversarial perturbations hide 23 of 24 detected attacks from an LSTM-based IDS in the continuous regime, at an average cost of perturbing 2.87 of 12 sensors.

Mitigating the semantic integrity problem needs four capabilities:

(a) *Production intent modeling*: translating engineering flows and business processes into formal, machine-checkable safety specifications.
(b) *Trace-level anomaly detection*: checking whole command sequences, not individual packets, against intent specifications.
(c) *Process mining*: converting raw device registers into event logs from which behavioral models, and deviations from them, can be discovered [21].
(d) *Constraint propagation gating*: verifying that a multi-step command sequence satisfies global process invariants before any command in it reaches an actuator.

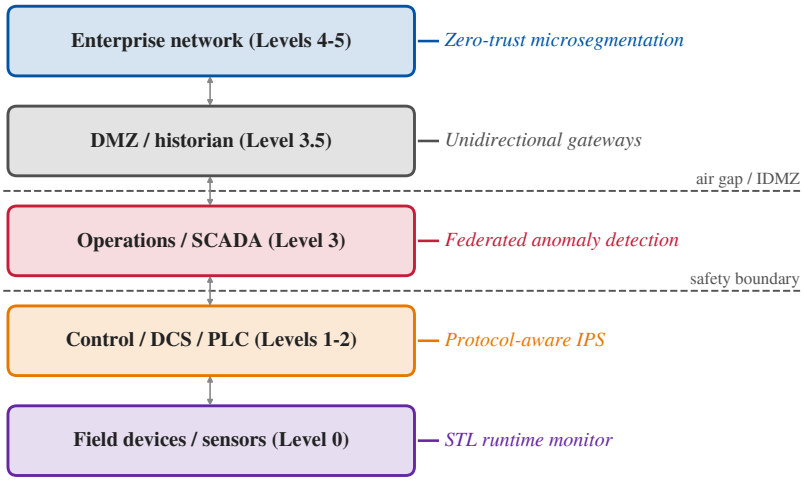



**Fig. 12** OT security architecture combining Purdue Model zone segmentation with zero-trust microsegmentation, federated anomaly detection, protocol-aware intrusion prevention, and STL runtime monitoring at each level. Dashed lines denote the industrial demilitarized zone (IDMZ) and the safety boundary.

### 6.5.3 Zero-Trust Microsegmentation at the IT/OT Convergence Seam

Zero-trust microsegmentation enforces fine-grained, per-asset access policies inside isolated OT zones. CISA, NIST, and the Department of Defense all name it as the main architectural control for securing the point where IT and OT meet [139]. CISA's Zero Trust Maturity Model maps these requirements onto mixed OT, ICS, and IoT estates and uses microsegmentation to limit lateral movement, and the DoD applies the same principles to isolating autonomous control assets.

Three operational constraints shape how microsegmentation can be deployed in industrial settings. Most OT endpoints run on legacy microcontrollers that cannot host security agents or manage certificates, so segmentation has to be enforced *externally* by network enforcement nodes placed on the communication lines [140]. Continuous-process plants cannot tolerate downtime, so policies are rolled out in stages, with traffic first modeled and confirmed and only then blocked [141]. And IP- or port-level filtering is not enough for industrial workflows: enforcement nodes have to parse legacy protocols such as Modbus, DNP3, and PROFINET to tell legitimate commands from anomalous parameter changes.

Fig. 12 shows an OT security architecture that combines Purdue Model zone isolation with zero-trust microsegmentation and edge anomaly detection at every level.

**Table 13** ICS/OT protocol security properties and documented attack vectors

| Protocol | Auth | Encrypt | Integrity | Known Documented Attack Vectors |
|---|---|---|---|---|
| Modbus | ✗ | ✗ | ✗ | Packet replay, direct command injection |
| DNP3 | Optional | ✗ | CRC Only | Station spoofing, man-in-the-middle injection |
| EtherNet/IP | ✗ | ✗ | ✗ | Session hijacking, network denial-of-service |
| PROFINET | ✗ | ✗ | Frame CRC | EtherType spoofing, ARP cache manipulation |
| OPC-UA | ✓ | TLS | ✓ | Trust certificate exploitation, endpoint flooding |
| IEC 61850 | ✓ | Optional | ✓ | MMS packet replay, GOOSE storm flooding |

✓= fully supported via protocol specifications; ✗= absent by design.

### 6.5.4 Protocol-Level Vulnerabilities Across the Communication Plane

Legacy ICS protocols (Modbus, DNP3, EtherNet/IP, PROFINET) were designed without security features, and OPC-UA added them later. Table 13 summarizes the security properties of each protocol and the attack vectors documented against it.

Four of the six protocols in the table have no native authentication and no encryption, and they rely on CRC checks that an adversary can forge easily. Modbus, deployed on hundreds of millions of endpoints, has no native security at all [38, 44]. DNP3 offers optional authentication but no native encryption, so even authenticated command streams travel in plaintext and can be read by a passive eavesdropper.

OPC-UA and IEC 61850 are the exceptions. OPC-UA uses TLS for client-to-server encryption and certificate-based authentication, and IEC 61850 enforces authentication with optional encryption profiles. Many plants nevertheless stay on insecure protocol versions because migration risks process disruption or added latency [37].

Protocol-agnostic traffic profiling with machine learning has therefore become an important compensating control. Nader et al. [43] apply $\ell_p$-norm one-class classifiers to SCADA traffic and reach 97% detection accuracy with few false positives on simulated anomalies. Kravchik and Shabtai [40] combine lightweight neural networks with principal component analysis (PCA) for similar accuracy at a third of the computational cost, which makes edge deployment in OT plausible. Li et al.'s *DeepFed* [42] aggregates anomaly detection models across ICS sites without sharing raw telemetry, which suits operators who cannot let process data leave the plant.

Semantic attacks on legacy protocols deserve particular attention. Cardenas et al. [36] showed that an attacker with process knowledge can issue Modbus register writes that each stay within normal limits and together drive the system into a dangerous state; this is the canonical exploit behind Fig. 11. Stuxnet [35] is the real-world example: its Siemens S7 PLC commands were structurally valid but sequenced to destroy centrifuges over several weeks. Anomaly detectors that analyze whole command traces rather than single packets [20] are the main defense, and they are still mostly research prototypes.

**Table 14** Reported performance of AI-based OT anomaly detection methods. All $F1$ ranges and false-positive rates come from testbeds with static, human-scripted attacks. Whether they hold against sequences generated by reinforcement learning, or against stealthy wear-out attacks, is not known.

| **Method** | Technique | Dataset | F1 (%) | FPR (%) | Adv. Robust. | Latency | Fed. | Key Limitation |
|---|---|---|---|---|---|---|---|---|
| **CUSUM/ EWMA (SPC)** | Statistical control | SWaT, HAI | 72–81 | 8–15 | Low | <1 ms | No | Structured attacks evade bounds |
| **Random Forest** | Supervised ML | SWaT, BATADAL | 88–94 | 3–8 | Medium | 5–20 ms | Partial | Retraining needed for concept drift |
| **LSTM/ GRU** | Deep learning | SWaT, BATADAL | 90–97 | 2–5 | Low [8] | 20–80 ms | Yes | Adversarial evasion |
| **Autoencoder** | Unsupervised | HAI, IIoT | 85–93 | 4–9 | Low | 10–50 ms | Yes | Threshold sensitivity |
| **Graph neural network** | Topology-aware | SWaT | 91–96 | 2–6 | Medium | 30–100 ms | Partial | Requires topology ground truth |
| **Federated ensembles [15]** | Multi-site FL | Multi-site ICS | 88–94 | 3–7 | Medium | 40–120 ms | Yes | Aggregation-stage poisoning |

FPR = false positive rate. Adv. = adversarial. Fed. = federated-learning capable. F1 ranges are min–max across studies in the corpus; they are *not* confidence intervals. All figures are upper bounds obtained under idealized, rule-bound threat conditions (cf. Section 9).

### 6.5.5 Intelligence-Driven Anomaly Detection and Model Limitations

Recent threat reports describe a move from reactive signature matching to adaptive, intelligence-driven anomaly detection in OT [119, 120], driven by the failure of rule-based filters against new cross-plane attacks. Table 14 extends the comparison of Table 10 with OT-specific methods and federated architectures.

Beyond the methods in the table, four families of ML deployment recur in the OT literature. Autoencoders trained only on clean process-variable histories flag deviations through spikes in reconstruction error; the *EvoAAE* architecture uses adversarial autoencoders and keeps detection rates high on complex IIoT distributions, including zero-day attack traces. Association rule mining extracts normal operational dependencies and timing from historical data logs and flags out-of-order execution. Graph neural networks represent the plant's communication fabric as a dynamic graph and detect small communication deviations and topology poisoning. Multi-site federated learning trains detectors across independent industrial sites and aggregates model weights without moving local data [15].

## 6.6 Layer 5: Knowledge and Feedback

### 6.6.1 Architecture and Information Flows

Layer 5 is where the PAIS taxonomy learns. It receives evidence from all other layers, maintains the invariant registry that L2 and L3 read, and generates red-team scenarios

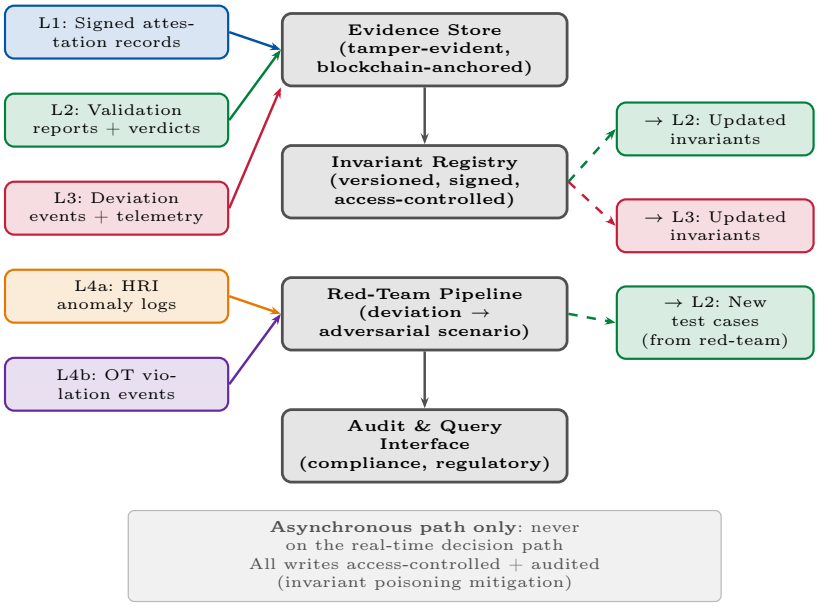



**Fig. 13** Knowledge & Feedback layer (L5) architecture. All inputs from Layers 1–4 flow into the Evidence Store. The Invariant Registry outputs to L2 and L3. The Red-Team Pipeline generates adversarial test cases fed back to L2 validation. All paths are asynchronous and access-controlled.

from observed deviations. L5 is *asynchronous*: it is never on the real-time decision path. Fig. 13 shows its information flows.

### 6.6.2 MITRE ATLAS as a Knowledge Infrastructure

MITRE ATLAS [124] provides a structured knowledge base of adversarial ML techniques. The Physical AI knowledge layer should both use it and contribute to it. As of the v5.1.0 data release (November 2025), ATLAS covered 16 tactics, 84 techniques, 56 sub-techniques, 32 mitigations, and 42 real-world case studies documenting AI security incidents, and later monthly releases have continued to grow it [124].

Physical AI security incidents not represented as ATLAS case studies as of that release include: prompt injection on LLM-integrated robots [131], hardware trojan activation in physical actuator control stacks [136], and DRL wear-out attacks on ICS [10]. L5 evidence stores are well placed to supply these cases. When a Physical AI system confirms an attack, its signed, structured L5 records are already the documentation an ATLAS case study needs, and contributing them feeds operational experience back to the adversarial ML research community.

### 6.6.3 Tamper-Evident Evidence Stores

Physical AI decisions have legal, safety, and compliance implications, so the evidence store has three requirements.

- *Tamper evidence*: any modification to records must be detectable. Blockchain-based approaches [73] provide cryptographic tamper evidence, but their throughput and latency costs are too high for real-time Physical AI decision paths. Hash-chained append-only logs [72] offer a lighter-weight alternative with tamper-evidence properties suitable for high-frequency telemetry.
- *End-to-end signing*: every record should carry cryptographic provenance from the originating layer (attestation record, validation report, deviation event). L1 attestation keys [47] provide the root signing identity; records signed with TPM-backed keys cannot be repudiated by the originating device.
- *Auditability*: compliance reviewers (safety, security, regulatory) need structured query interfaces. The EU AI Act [86] Article 12 requires logging for high-risk AI systems, and for Physical AI the evidence store's query interface is how that requirement is met.

Mishra et al. [29] implement a lightweight, blockchain-inspired SHA-256 hash-chained ledger for trust-score updates in an IIoT intrusion-detection framework and report negligible added latency; because it omits distributed consensus, it provides tamper evidence rather than distributed trust anchoring. Rathee et al. [74] propose a blockchain-based secure communication framework for industrial IoT that demonstrates practical feasibility in a cobot manufacturing scenario. Ali and Mahrawy [73] survey blockchain applications for cybersecurity across CPS domains and find that data integrity and provenance are the main benefits, which are also the two things an L5 evidence store needs.

### 6.6.4 Invariant Registry Management

The invariant registry is the most security-sensitive component of L5. A successfully poisoned invariant registry can cause L2 to admit physically unsafe actions (if safety bounds are weakened) or cause L3 to miss deviations (if detection thresholds are raised). It needs three mitigations:

1. *Write access control*: only authorized domain experts (with role-based access control verified against L1 attestation) can add or modify invariants.
2. *Cryptographic versioning*: every invariant version is signed; L2 and L3 reject unsigned or version-downgraded invariants. This gives an *invariant freshness* guarantee and blocks replay attacks that roll the registry back to a less restrictive version.
3. *Conflict detection*: when new invariants conflict with existing ones (e.g., relaxing a safety bound that another invariant tightens), the conflict must be flagged and resolved before the registry update is published. STL [81] provides a formal framework for detecting semantic conflicts between temporal invariants, though automated conflict detection for spatial and semantic invariant types remains open.

Process mining tools [76] can be used to *discover* candidate invariants from historical operational logs, which domain experts then confirm or refine. This eases the cold-start problem of filling an empty invariant registry for a new deployment.

### 6.6.5 Red-Team Scenario Pipeline

The red-team scenario pipeline closes the learning loop by turning confirmed runtime deviations into adversarial test cases for L2 pre-deployment validation. This gives the defender an asymmetric advantage. Every attack that succeeds at runtime becomes a permanent test case in future validation, so the adversary needs a new attack each time while the defender's corpus of validated attack scenarios keeps growing.

Three recent approaches to automated scenario generation at L5 are relevant to Physical AI:

(1) *LLM-assisted scenario generation*: PentestGPT [12] splits LLM reasoning, generation, and parsing into separate modules and iterates to find exploit paths that human testers might miss. Adapted to Physical AI and fed with deviation logs, it could generate OT-specific adversarial scenarios from incident reports and MITRE ATT&CK techniques.
(2) *DRL adversarial scenario generation*: the DRL techniques behind wear-out attacks [10] can also generate training scenarios for defensive anomaly detectors, giving a red team that learns the minimal perturbations needed to evade the current detection configuration.
(3) *Curriculum learning for continuous hardening*: curriculum learning scores the difficulty of detected anomalies and feeds progressively harder examples back into the validation twins (L2) and edge detectors (L3). This improves detection of zero-day threats *without catastrophic forgetting* [162], which matters for Physical AI detectors that must keep covering known attacks while learning new ones.

The pipeline must be protected against its own poisoning: if an adversary can inject malicious scenario descriptions into the pipeline, they could cause L2 to approve unsafe actions it previously rejected. Access control on the pipeline is therefore as important as access control on the invariant registry, and cryptographic signing of scenarios is required at the pipeline boundary.

# 7 End-to-End PAIS Reference Architecture

*Status of this section.* The reference architecture that follows is a *design pattern* synthesized from the gaps identified in the preceding sections. It has not been implemented, instantiated, or measured. No prototype exists, no latency figure reported here was obtained from a running system, and no security property claimed for the composition has been empirically validated. We label it as such throughout and avoid evaluative language. Its purpose is to make the cross-layer seams explicit enough to be argued with and built against; the evaluation agenda that would move it from pattern to validated architecture is stated in Section 7.4.

Section 6 showed that each PAIS layer has an active literature whose mechanisms are developed in isolation, and Section 8 shows that no established framework spans the complete AI-to-actuation lifecycle either (Table 18). This section connects the security mechanisms identified across the five layers along the AI-driven decision lifecycle, so that hardware trust, validation, runtime assurance, semantic checks, evidence

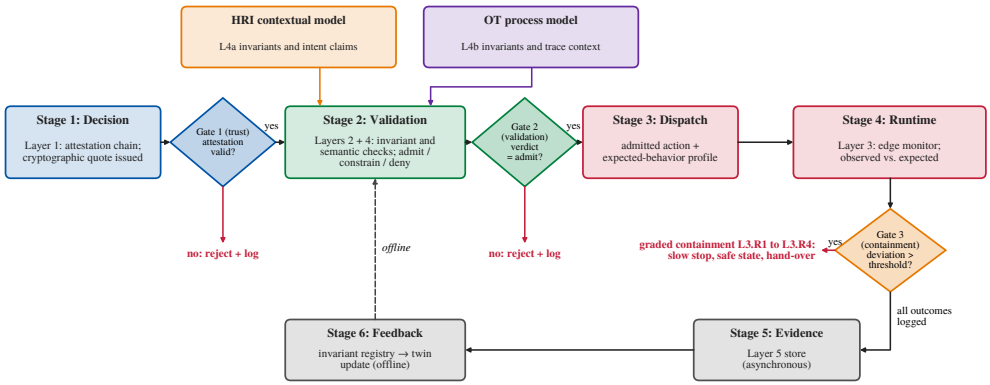



**Fig. 14** End-to-end decision lifecycle of the reference pattern. Gates 1 and 2 are hard gates: a rejected or unauthenticated decision is dropped before it reaches a plant actuator. Gate 3 is a soft, graded local containment trigger. The HRI and OT semantic layers supply domain invariants to the validation stage, and Stage 6 runs the asynchronous feedback loop.

capture, and feedback interact through explicit interfaces rather than by assumption. Table 15 lists, for each stage, the surveyed mechanisms it draws on and the artifact it passes to the next stage.

## 7.1 Synthesized End-to-End Reference Workflow

The workflow below traces the path an AI-driven decision takes, through a series of verification states, before actuation is permitted at the edge. It has six sequential stages spread across the five layers, each guarded by its own verification gate, and Fig. 14 shows it end to end.

1. *Stage 1, decision generation (L1).* The edge platform runs neural network inference and binds the output to a signed Platform Configuration Register (PCR) attestation token that vouches for the integrity of the model weights and the firmware stack.
2. *Stage 2, invariant and semantic validation (L2, L4).* The attested decision reaches the first gate. L2 runs the proposed trajectory through a behavioral twin to check that no plant safety rule or physical envelope limit is violated. L4 semantic checks are applied at the same gate. For a command from a human, L4a checks that it is an authenticated, uncoerced expression of operator intent (multimodal congruence, plus dual-operator consensus for high-consequence actions). For an OT command, L4b checks it against the recent command trace and rejects a sequence of individually valid commands that violates production intent (Section 6.5).
3. *Stage 3, dispatch with an expected profile (L3).* Each admitted command is tagged with the operational profile it is expected to produce, for example motor torque thresholds, joint velocity expectations, and network telemetry bounds.
4. *Stage 4, runtime monitoring (L3).* Edge monitors with a sub-50 ms budget compare live sensor telemetry with the profile issued at Stage 3.

**Table 15** Reference architecture stages mapped to the layer mechanisms surveyed in Section 6 and to the artifact each stage passes on. Seams A–C are defined in Section 7.3.

| Stage | Layer(s) | Surveyed mechanisms connected at this stage | Artifact passed on |
|---|---|---|---|
| **1. Decision generation** | L1 | TPM 2.0 / firmware-TPM quotes, TEEs, secure boot, model-hash binding in sealed PCRs, amortized and cached attestation (Section 6.1) | Verified-decision envelope (Seam A) |
| **2. Invariant and semantic validation** | L2, L4a, L4b | Behavioral digital twin with invariant checking (STL/MTL); offline formal verification of the controller; multi-modal intent verification and dual-operator consensus; trace-level semantic integrity checking (Sections 6.2, 6.4, 6.5) | Admit / constrain / deny verdict |
| **3. Dispatch with expected profile** | L3 | Expected multi-modal behavior profile synthesized at decision time | Validation report and profile (Seam B) |
| **4. Runtime monitoring and containment** | L3 | Hybrid invariant-rule and ML edge detectors, STL hardware monitors, graded containment hierarchy L3.R1–R4, control barrier functions (Section 6.3) | Containment action; deviation events |
| **5. Evidence capture** | L5 | Hash-chained append-only logs or blockchain anchoring; TPM-backed record signing (Section 6.6) | Signed evidence records |
| **6. Registry tuning and red teaming** | L5 | Signed, versioned invariant registry with conflict detection; LLM-assisted, DRL-based, and curriculum-driven scenario generation (Section 6.6) | Updated invariants (Seam C); new L2 test cases |

5. *Stage 5, evidence capture (L5).* Attestation logs, twin verdicts, HRI anomaly logs, OT violation events, and runtime telemetry traces are signed and written to an append-only ledger.
6. *Stage 6, registry tuning (L5).* Out of band, a learning pipeline processes the logged deviations and pushes signed, updated invariants back into the L2 validation registry.

## 7.2 Cross-Layer Dependencies and Failure Modes

No single layer is sufficient on its own, although each is needed. Fig. 15 shows the dependencies between layers.

The dependencies are the argument for defense in depth: each layer can contain some of what escapes from its neighbors. An attacker who defeats Layer 1 (with a hardware Trojan that extracts keys, say) can issue decisions with valid attestation headers, but the Layer 2 twin should still reject the trajectory on physical-safety grounds. An attacker who defeats Layer 2 (by poisoning the validation twin, for instance) still has to get past the Layer 3 edge monitors. Whether this redundancy holds in practice is exactly what experiment E3 below would test.

Three cascading failures are the most damaging:

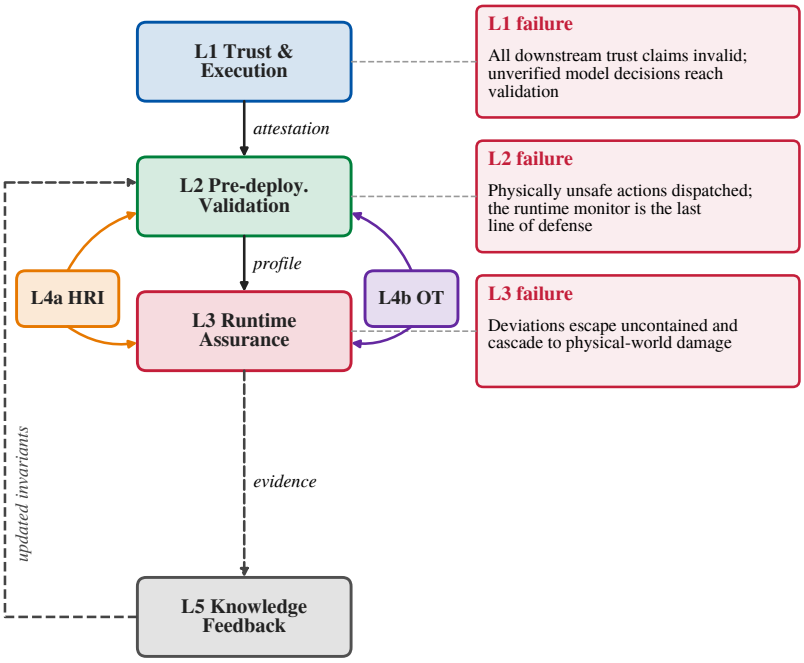



**Fig. 15** Cross-layer dependencies between the PAIS layers. When a layer fails, the residual risk passes downward toward the physical execution loop.

- *L1 attestation cascade.* If the hardware root of trust is compromised, every downstream attestation claim is void. A Layer 2 node that accepts a forged attestation token will validate decisions made with tainted weights, and the Layer 3 edge loop becomes the last line of physical containment.
- *L2 validation cascade.* If the verification twin is poisoned, or an insider manipulates the invariant registry, hazardous actions are admitted. Layer 3 is then the only protection left, and if its detection thresholds come from the same poisoned registry, it will miss the deviation too.
- *L5 feedback cascade.* If the invariant registry or the red-team pipeline is poisoned, the out-of-band feedback loop becomes an attack vector. The adversary can relax safety envelopes a little at a time and blind Layer 2 and Layer 3 together.

## 7.3 Contract-Based Integration and Latency Budgets

The pattern decouples layers through *published, versioned contracts* rather than shared code, so that a component can be replaced, maintained, or verified on its own. It has three seams:

- *Seam A, trust contract (L1 ↔ L2).* L1 emits a *verified-decision envelope* that bundles the candidate action, the hardware attestation quote, and the stack's integrity measurements. L2 drops any envelope that fails cryptographic validation. The

envelope schema is versioned so that firmware attestation can change without a simultaneous rewrite of the L2 validator.

- *Seam B, verdict-and-profile contract (L2 ↔ L3).* L2 emits a *validation report* with the approved action and its expected physical behavior profile, which L3 treats as ground truth for runtime checking. The profile has to be fresh: for dynamic systems whose expected trajectory changes with every inference, it must be generated at decision time rather than cached.
- *Seam C, invariant contract (L4 ↔ L2, L3).* A versioned, access-controlled registry schema records each invariant's subject, predicate, scope, severity, and an evaluation hint. Domain experts can extend plant security policy without changing the twin or monitor code, and the evaluation hint tells the L2 twin and L3 monitor which checker to use for that invariant (an STL model checker, a hard-coded SIL safety rule, or a GNN trace classifier, for example).

The assume-guarantee (A-G) contract framework of Seshia et al. [103] and Bartocci et al. [104] is the theoretical basis for this decoupling. In A-G terms, each layer guarantees a property of its outputs provided that its assumptions about its inputs hold. Seam A's envelope carries Layer 1's identity guarantee, conditional on the attestation chain being intact. Seam B's report carries Layer 2's safety guarantee, conditional on the invariants being complete and unpoisoned. Seam C's registry carries Layer 4's domain-correctness guarantee, conditional on access control over the registry.

The conditions are part of the guarantee. A compromised Layer 1 voids Seam A, and incomplete plant invariants weaken Seam B. The pattern offers no unconditional security proof. What it offers is a structure in which the assumptions, guarantees, and residual risks at each seam are written down, so that engineers can analyze how the layers compose.

Allocating the available latency budget across these seams is the binding systems-engineering constraint on the pattern. We state it as a set of *design targets*, not as a measured result: no instantiation of this architecture has been built or profiled, and the component intervals below are indicative ranges taken from the mechanisms surveyed in Sections 6.1–6.5, not from an implementation.

Consider a control loop with a 100 ms decision-to-actuation budget, and write the per-seam contributions as intervals:

$$\begin{aligned} T_{\text{total}} &= T_{\text{L1}} + T_{\text{L2}} + T_{\text{L3}} + T_{\text{L4}}, \\ T_{\text{L1}} &\in [5, 11], \quad T_{\text{L2}} \in [30, 80], \\ T_{\text{L3}} &\in [1, 20], \quad T_{\text{L4}} \in [5, 15] \quad \text{(ms)}. \end{aligned} \tag{2}$$

The nominal-case sum is 41 ms; the worst-case sum is 126 ms. The 100 ms budget is therefore *not* satisfied at the worst case, and the partition cannot be presented as a satisfied inequality. Two consequences follow, and they are the useful content of the analysis.

*First, feasibility is a constraint on L2, not a property of the stack.* Holding the other three seams at their worst case, the budget admits

$$T_{\mathrm{L2}} \;\leq\; 100 - (11 + 20 + 15) \;=\; 54 \text{ ms}. \tag{3}$$

A behavioral-twin evaluation that can exceed 54 ms cannot occupy the synchronous path at all. It must instead be restructured: pre-computed over an admissible action set, evaluated speculatively ahead of the decision, or moved off the critical path with L3 explicitly carrying the residual risk. Which of these is acceptable is a safety-case question, not a performance-tuning question.

*Second, the L1 term cannot be a per-decision cost.* As noted in Table 6, a TPM 2.0 quote is a multi-millisecond operation and is substantially slower on discrete parts [155, 156]; per-decision attestation would consume a large share of the budget by itself. L1 attestation must therefore be amortized across decisions as a cached, periodically refreshed state whose validity window is itself a security parameter, traded against the staleness an attacker could exploit within it.

Both conclusions rule out cloud round-trips and remote policy enforcement inside the control loop. Both are hypotheses to be tested by the instantiation called for in Research Challenge R6, not measurements reported by this survey.

## 7.4 Evaluation Agenda for the Reference Pattern

Because the pattern is unevaluated, we state what would constitute evaluating it, so that the claim can be falsified rather than merely adopted. Four experiments are required, and they are the concrete content of Research Challenges R6 and R7.

*E1, instantiation and stage profiling.* Build the six-stage workflow on two substrates with different failure semantics: a collaborative robot cell (L4a-dominant) and a SWaT-class process testbed (L4b-dominant). Report measured per-stage latencies for Stages 1–6 against the intervals assumed in Eq. (2), and report the attained $T_{\mathrm{L2}}$ against the 54 ms constraint derived above. The pattern is falsified as a real-time architecture if $T_{\mathrm{L2}}$ cannot be brought under that bound without moving the twin off the synchronous path.

*E2, seam-contract conformance.* Verify that Seams A, B, and C can be versioned and substituted independently, by replacing one layer's implementation (for example, TPM-based for TrustZone-based attestation at L1) without recompiling its consumers. The decoupling claim fails if substitution requires coordinated change across seams.

*E3, cascading-failure validation.* Inject the three failure cascades of Section 7 (L1 attestation compromise, L2 twin poisoning, L5 registry poisoning) and measure whether the surviving layers contain the fault as the assume-guarantee analysis predicts. This is the only experiment that tests the defense-in-depth claim rather than assuming it.

*E4, adversarial evaluation under AI-native attack.* Because the benchmarks available today contain only scripted attacks (Section 3.6), E1–E3 must be repeated under DRL-crafted wear-out sequences and LLM-generated command sequences, which requires the hardware-in-the-loop infrastructure of R7. Until E4 is performed, no claim about the pattern's resistance to AI-native adversaries is supportable.

We report no results for E1–E4 and make no claim as to their outcome.

# 8 Security Framework, Standards, and Regulatory Mapping

This section assesses how the PAIS taxonomy relates to three families of established instruments: cybersecurity knowledge bases (Section 8.1), security frameworks and reference architectures (Section 8.2), and industrial security standards, robotics safety standards, and AI regulation (Sections 8.3 and 8.4). For each family we identify where coverage exists, where it is fragmented or incomplete, and where Physical AI introduces cross-domain requirements that current instruments do not represent. Section 8.5 consolidates these findings.

## 8.1 Alignment with MITRE Knowledge Bases

We first align the PAIS layers with the two MITRE frameworks that describe adversary behavior, ATT&CK for ICS and ATLAS, so that each layer can be traced to concrete threat intelligence.

MITRE ATT&CK for ICS [125] defines 11 tactics applicable to industrial environments. Srinivasan and Karimi [130] find that OT-specific attack compositions remain under-represented in ATT&CK relative to their operational frequency. Table 16 maps all 11 ATT&CK ICS tactics to PAIS layers.

MITRE ATLAS (Adversarial Threat Landscape for AI Systems) [124] is the main community knowledge base for attacks on ML systems. As of the v5.1.0 data release (November 2025), ATLAS documents 16 tactics, 84 techniques, and 56 sub-techniques covering the adversarial ML lifecycle, and monthly releases since then have extended it, predominantly with agent-focused techniques [124]. Separately, and as context rather than as a Physical AI-specific measurement, CrowdStrike reports an 89% year-on-year increase in attacks by AI-enabled adversaries during 2025 across *all* domains it observes [148]. ATLAS has no Physical AI profile that maps its techniques to physical consequence classes or to OT/ICS environments. Connecting ATLAS to the PAIS taxonomy would mean documenting Physical AI case studies such as prompt injection against LLM-driven robots, DRL wear-out attacks, and hardware Trojan activation; the Layer 5 evidence stores of Section 6.6 are a natural source for them (Fig. 16).

That alignment covers adversary behavior but not defensive countermeasures, embedded-device threats, or weakness-level traceability. A taxonomy built around defense layers needs those as well, so we extend the mapping to three further MITRE knowledge bases.

MITRE D3FEND [126] is the defensive mirror of ATT&CK: a knowledge graph of countermeasures organized into seven tactics (Model, Harden, Detect, Isolate, Deceive, Evict, Restore), which reached v1.0 in January 2025 and, in v1.3.0 (December 2025), comprises 267 defensive techniques and ships an explicit operational-technology extension. The OT extension makes D3FEND directly applicable to L3 and L4b, not just analogous to them. MITRE EMB3D [127], publicly released in May 2024, models threats to embedded devices, maps them to device properties, and aligns to CWE, CVE, and ATT&CK. It is the obvious threat model to pair with PAIS Layer 1, which

**Table 16** MITRE ATT&CK for ICS: 11 tactics mapped to PAIS layers

| ATT&CK Tactic | PAIS | Representative Technique |
|---|---|---|
| Initial Access | L1 | Exploit public-facing SCADA (T0819) |
| Execution | L3 | Modify control logic (T0833) |
| Persistence | L1 | Module firmware (T0839) |
| Privilege Escalation | L1/L3 | Exploitation for privilege (T0890) |
| Evasion | L3 | Spoof reporting message (T0856) |
| Discovery | L3 | Remote system info. (T0888) |
| Lateral Movement | L3 | Remote services (T0886) |
| Collection | L4b/L5[‡] | Tag identification (T0861) |
| Command & Control | L3[§] | Standard application protocol (T0869) |
| Inhibit Response | L4b | Firmware update mode (T0800) |
| Impair Process Ctrl | L4b | Modify parameter (T0836) |

[‡] Tag identification is an OT-semantic reconnaissance activity against the process model and is therefore assigned to L4b; assigning it to L5 on the grounds that its product is knowledge would conflate the attacker's knowledge with the defender's evidence store. L5 is a secondary assignment only because tag enumeration is also visible in evidence-store telemetry.
[§]Retained at L3, with the justification made explicit: PAIS layers denote the *defensive control point*, not the attacker's objective. C2 over a standard application protocol is observable and interdictable at the runtime-assurance boundary, which is where the corresponding defense sits; no claim is made that C2 is semantically a runtime-assurance activity.

is about embedded trust anchors. CAPEC and CWE [128, 129] supply attack-pattern and weakness-level anchors that let a PAIS layer be traced to concrete implementation defects. Finally, MITRE Engage, which superseded MITRE Shield as the adversary-engagement and deception framework, is the appropriate home for the red-team and deception elements of Layer 5, although we note that no Engage activity is currently specialized to cyber-physical deception.

Table 17 gives the layer-level mapping.

## 8.2 Coverage by Security Frameworks and Reference Architectures

Table 18 maps established security frameworks and reference architectures onto the PAIS layers. Individual standards and research frameworks each solve part of the problem, but none of them covers the whole operational lifecycle.

Table 18 shows three gaps. First, *no peer-reviewed framework covers the whole AI-to-actuation lifecycle.* Each existing framework handles one part: hardware trust boundaries (OpenTitan), network anomaly monitoring (IEC 62443, Alias Robotics RIS), or descriptive attack matrices (MITRE ATT&CK ICS). None publishes integration contracts that span the full decision path.

Second, *human–robot interaction security is left out.* ISO 10218-1/2:2025 now has safety-related cybersecurity requirements, but they are scoped to protecting the integrity of safety functions. Together with the device patching of Alias Robotics

**Table 17** Extended MITRE alignment: PAIS layers mapped to D3FEND countermeasure classes, EMB3D applicability, and representative CAPEC/CWE anchors. CAPEC and CWE entries are *representative*, not exhaustive; the complete per-threat mapping for all 19 threat types of Table 5.

| PAIS layer | D3FEND countermeasure classes | EMB3D applicability | Representative CAPEC / CWE | Residual gap |
|---|---|---|---|---|
| **L1 Trust & Execution** | Harden: platform hardening, bootloader authentication, TPM boot integrity; Model: asset inventory | Direct: bootloader, firmware, and hardware threat categories map one-to-one | CAPEC-441 (malicious logic insertion); CAPEC-186 (malicious software update); CWE-1326 (missing immutable root of trust) | No countermeasure addresses *model* attestation as distinct from platform attestation |
| **L2 Predeploy. Validation** | Model: operational activity mapping; Harden: application hardening | Partial: device-level only, no model-behavior analogue | CWE-1039 (inadequate handling of adversarial input perturbations) | No D3FEND or EMB3D construct for behavioral-twin or invariant-based admission control |
| **L3 Runtime Assurance** | Detect: network and process analysis; Isolate: execution and network isolation; *OT extension applies* | Partial: device-resident detection | CAPEC-94 (adversary in the middle); CWE-294 (capture-replay bypass) | Detection analytics assume IT-rate telemetry, not sub-50 ms control loops |
| **L4a HRI Semantics** | *None applicable* | ✗ Not in scope | CAPEC-416 (manipulate human behavior) | No MITRE knowledge base models the operator as an attack surface; the L4a gap recurs in all of them |
| **L4b OT Semantics** | Detect: process and message analysis; Isolate: broadcast and execution isolation; *OT extension applies* | Partial via ATT&CK for ICS linkage | CWE-306 (missing authentication for critical function) | No construct expresses multi-command *sequence* semantics |
| **L5 Knowledge & Feedback** | Restore and Evict tactics; Deceive (and MITRE Engage) for red-team and deception pipelines | ✗ Not in scope | CWE-778 (insufficient logging); CWE-117 (improper log neutralization) | No framework treats the security knowledge store itself as a poisoning target |

The L4a row is the finding, not an omission: the absence of operator-directed countermeasures across ATT&CK, ATLAS, D3FEND, and EMB3D independently corroborates the corpus-based L4a sparsity result of Section 9.

RIS, they leave Layer 4a without any framework that can deal with deliberate cognitive manipulation or authority spoofing. Third, *hardware-centric frameworks have no knowledge feedback*. A standalone root of trust such as OpenTitan has no standard path for taking in runtime anomalies or validation reports, so it cannot re-attest or update its security profile out of band. Section 7 proposes a reference architecture that addresses these gaps; its row in Part B of Table 18 records intended, not demonstrated, coverage, and its empirical validation is Research Challenge R6.

## 8.3 Standards and Regulatory Coverage

Physical AI falls under three regulatory traditions: industrial safety, cybersecurity, and AI-specific regulation. Table 19 lists the main instruments, and Fig. 17 shows their coverage of the PAIS layers as a compliance heatmap.

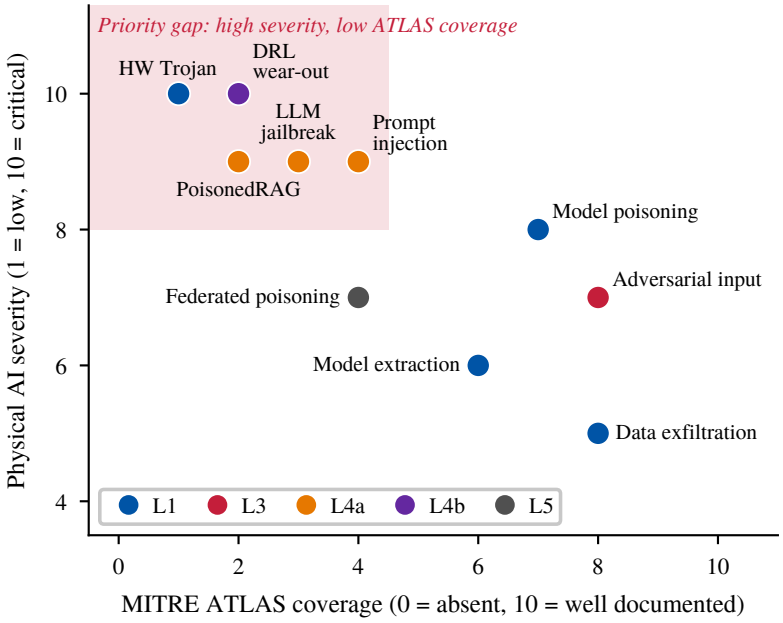



**Fig. 16** MITRE ATLAS coverage against Physical AI operational severity (ordinal author scores); color gives the primary PAIS layer. The shaded region is the priority gap (high severity, low ATLAS coverage). Hardware Trojans, DRL wear-out attacks, and PoisonedRAG are the most urgent Physical AI gaps: each is weakly covered by ATLAS, and none was documented there as a Physical AI case study as of the v5.1.0 release.

Three patterns stand out in the heatmap. First, L4a (HRI security) is the least covered layer: six of the ten instruments provide no coverage, and none addresses it fully; the four that provide partial coverage (ISO 10218-1/2:2025, ISO/TS 15066, ISO 13849, and the EU AI Act through its human-oversight provision) treat the human–robot interface as a matter of safety or oversight rather than security. Second, L3 (Runtime Assurance) is the best-covered layer, addressed by eight of the ten instruments, which matches the maturity of the OT literature on anomaly detection, intrusion detection, and incident response. Third, L1 (trust) receives only partial coverage from most instruments, and none mandates hardware attestation for AI inference nodes, which is the central gap for Physical AI trust. The full clause-level justification for all non-zero cells (e.g., IEC 62443-3-3 SR 6.2, *Continuous monitoring*, under FR 6 *Timely response to events*, for L3 coverage; EU AI Act Art. 15, *Accuracy, robustness and cybersecurity*, for L3 and L1; ISO 10218-1:2025 cybersecurity provisions for L4a partial coverage).

## 8.4 The EU AI Act and Physical AI

The EU AI Act (Regulation (EU) 2024/1689) classifies Physical AI systems as high-risk when used in critical infrastructure, industrial machinery, or systems making safety-relevant decisions. Its application is staged under Article 113 rather than arriving on a single date, and the timetable was amended by the Digital Omnibus on AI,

**Table 18** Security frameworks and reference architectures mapped to the PAIS taxonomy

| **Part A: Deployed and peer-reviewed security frameworks** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Architecture / Framework** | Domain | L1 Trust | L2 Valid. | L3 Run-time | L4 HRI/ OT | L5 Know. | Key Limitation |
| **IEC 62443** | OT/ICS | Partial | ✓ | ✓ | ✗ | Partial | No AI-specific trust; no HRI |
| **NIST CSF 2.0** | General | Partial | Partial | ✓ | ✗ | ✓ | Framework-level strategy only |
| **ROS 2 Security (SROS2)** | Robotics | ✗ | ✗ | Partial | ✗ | ✗ | Communication only; no model attestation |
| **MITRE ATT&CK ICS** | OT | ✗ | ✗ | Partial | ✗ | ✓ | Threat taxonomy only; no countermeasures |
| **OpenTitan** | HW Trust | ✓ | ✗ | ✗ | ✗ | ✗ | Confined strictly to hardware layer |
| **Alias Robotics RIS** | Robotics | Partial | ✗ | ✓ | Partial | Partial | Reactive patch-based; no formal validation |
| **ISO 10218-1/2:2025** | HRI Safety | Partial | Partial | ✓ | ✓ | ✗† | Safety-motivated cyber clauses; no adversary model |
| **Part B: Architectural pattern derived from corpus gaps** | | | | | | | |
| **Reference design pattern (this survey)** | Physical AI | ✓* | ✓* | ✓* | ✓* | ✓* | *Unevaluated*: intended coverage by construction, not demonstrated coverage |

✓=addressed per peer-reviewed evidence; ✗=not addressed; Partial=partially addressed. Research Challenge R6 covers cross-layer implementation.

†This table scores whether a framework itself *provides* a mechanism for each layer (an architectural lens), whereas the compliance heatmap of Fig. 17 scores clause-level *requirements*. ISO 10218-1/2:2025 supplies no knowledge-feedback mechanism (✗ here) but its logging and record-keeping clauses give it partial L5 coverage in the heatmap; the two lenses are complementary, not contradictory.

*Part B is a design pattern, not a deployed or evaluated system. Its cells denote coverage the pattern is *intended* to provide and are not commensurable with the evidence-based cells of Part A.

Regulation (EU) 2026/1744, in force since 27 July 2026 [163]. The distinction matters operationally for Physical AI: obligations for general-purpose AI models have applied since August 2025 and the Article 50 transparency obligations since 2 August 2026, whereas the high-risk obligations, originally due from 2 August 2026 (Annex III) and 2 August 2027 (Annex I), now apply from 2 December 2027 for stand-alone Annex III systems and from 2 August 2028 for high-risk systems that are safety components of products already covered by the Union harmonization legislation listed in Annex I, which under Article 6(1) is where most industrial robots and machinery fall. Physical AI deployers therefore face different deadlines depending on whether their system is regulated as a standalone Annex III high-risk system or as a safety component of a regulated product.

Compliance progress across the surveyed corpus is uneven: risk management and encrypted communications are comparatively well served, while explainability, real-time human oversight, and knowledge-base traceability remain thinly addressed. We report this ordering qualitatively: the corpus does not support a defensible denominator for proportions of solutions addressing each requirement.

The Act's requirements do not map one-to-one onto the PAIS layers. *Article 9* (risk management) maps to L2 pre-deployment validation; *Article 13* (transparency

**Table 19** Key standards and regulations for Physical AI security

| Standard / Regulation | Domain | Physical AI Relevance |
|---|---|---|
| IEC 62443-2-4 / 3-3 / 4-2 | OT/ICS | Part 2-4: SP requirements for IACS service providers; Part 3-3: System security requirements and security levels (SR 6.2 covers monitoring); Part 4-2: Technical security requirements for IACS components |
| NIST CSF 2.0 | General | Govern, Identify, Protect, Detect, Respond, Recover |
| NIST SP 800-82 Rev.3 | ICS | Guide to OT security; architecture guidelines |
| ISO 10218-1/2:2025 | Robotics | Industrial robot safety, robot classification, collaborative applications, and cybersecurity as it applies to robot safety |
| ISO/TS 15066:2016 | Robotics/HRI | Collaborative robot safety; PFL thresholds |
| ISO 13849 / IEC 62061 | Safety | Functional safety of machine control systems (PLd/SIL) |
| EU AI Act (2024) | AI | Risk-based AI regulation; high-risk categories include robotics and critical infrastructure |
| ISO/SAE 21434 | Automotive | Road vehicle cybersecurity engineering; applicable to AMRs |
| GDPR / DPDPA | Privacy | Personal data from robot sensors, HRI recordings |
| NERC CIP | Energy OT | Critical infrastructure protection for bulk electric systems |

and provision of information to deployers) maps to L5 audit interfaces; *Article 14* (human oversight) maps to L4a HRI security; and *Article 15*, whose subject matter is accuracy, robustness *and cybersecurity*, maps to L3 runtime assurance and, through its cybersecurity limb, to L1 trust. Article 15 is the Act's most direct textual hook for the PAIS trust and runtime layers and we rely on it in preference to weaker analogies. Conformity assessment is governed by *Article 43*, which selects between the procedures set out in *Annex VI* (internal control) and *Annex VII* (quality management system and technical documentation assessment); *Annex III* is a different instrument, enumerating the high-risk use-case categories that trigger those obligations rather than specifying the assessment itself. The conformity-assessment route therefore maps to L1 trust and model provenance and to L5 evidence retention jointly. An organization deploying Physical AI under the Act therefore cannot become compliant by working on one layer alone.

## 8.5 Synthesis: Covered, Fragmented, and Missing Coverage

Read together, the three families of instruments give a consistent picture. *Covered:* runtime assurance (L3) is represented in every family, through ATT&CK for ICS and the D3FEND OT extension (Tables 16 and 17), through IEC 62443 and NIST CSF 2.0 (Table 18), and through eight of the ten standards and regulations of Fig. 17; OT protocol and process security (L4b) is addressed by three instruments and partially

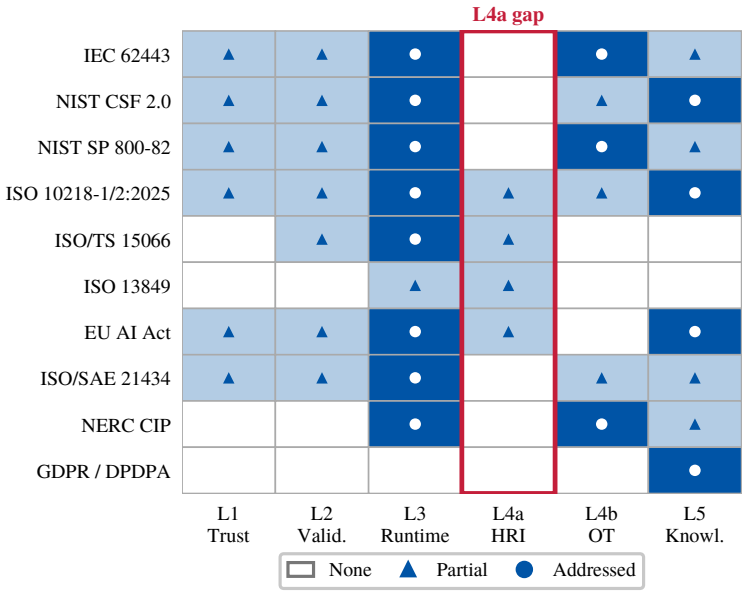



**Fig. 17** Regulatory compliance heatmap: 10 standards and regulations (rows) against 6 PAIS layers (columns). Light cells with ▲ = partial; dark cells with ● = addressed; white = none. The red box highlights L4a (HRI security), which remains uncovered by 6 of the 10 instruments. ISO 10218-1/2:2025 receives partial L1 coverage through its safety-related cybersecurity requirements, and the EU AI Act through the cybersecurity limb of Article 15. This heatmap scores clause-level requirements and therefore differs by lens, not in substance, from the architectural coverage of Table 18 (see its note †).

by three more. *Fragmented:* platform trust (L1), pre-deployment validation (L2), and knowledge feedback (L5) are each covered in parts that do not connect. EMB3D models embedded-platform threats and the D3FEND Harden tactic protects the platform, but neither addresses the model; no standard addresses L1 or L2 beyond partial coverage; and while evidence retention is required by the EU AI Act and NIST CSF 2.0, no instrument treats the knowledge store itself as a poisoning target. *Missing:* HRI security (L4a) is absent from every MITRE knowledge base, is covered in Table 18 only by the safety-motivated provisions of ISO 10218-1/2:2025 and partially by Alias Robotics RIS, neither of which models intentional manipulation, and receives at most partial, safety-motivated coverage from the other standards. Only ISO 10218-1/2:2025 has non-zero coverage on all six layers.

Physical AI also has three cross-domain requirements that no current instrument covers.

1. *AI-in-the-loop safety validation.* No standard specifies how to validate the safety of a neural network controller before deployment; ISO 10218 and IEC 62443 predate ML-driven control. ISO 13849 [89] provides performance level (PL) and safety integrity level (SIL) frameworks for safety-related control systems, but these are defined in terms of hardware and software failure rates. Those metrics are not

defined for neural networks, whose failure modes depend on the input distribution rather than on component reliability.

2. *Cross-domain invariant specification.* No standard provides an interoperable language for expressing cyber-physical invariants spanning robotics, OT, and HRI domains simultaneously. IEC 62443 [82] defines security levels for OT zones but does not specify the semantic content of invariants; ISO 10218 [84] specifies safety envelopes for robot arms but not the language for expressing them in machine-verifiable form.
3. *Hardware attestation for edge AI.* No standard mandates or specifies TPM-based attestation for edge AI inference nodes in industrial settings. The TCG TPM 2.0 specification [47] provides the technical foundation, and NIST SP 800-53 [88] includes remote attestation among its configuration management controls, but neither has been instantiated as a mandate for AI inference hardware in operational technology settings.

These gaps have legal consequences. The EU AI Act's conformity assessment requirements for high-risk AI systems (Article 43, following the procedures of Annexes VI and VII) require documented testing against safety, robustness, and cybersecurity criteria before market placement. For Physical AI deployments falling due under the Act's staged timetable, an operator who cannot demonstrate systematic pre-deployment validation (R1/R2 gap), runtime assurance (R3 gap), or hardware identity (attestation gap) may be non-compliant with legally binding requirements. The Act is therefore likely to push investment in Physical AI security.

Practical compliance requires bridging four standards families: IEC 62443 [82] (OT/ICS security), ISO 10218/15066 [84, 85] (robot safety), NIST CSF 2.0 [83] / SP 800-82 [37] (ICS cybersecurity guidance), and the EU AI Act [86] (AI regulation). None of these was designed with the others in mind, and their control frameworks use incompatible terminologies, risk models, and verification methodologies. The compliance heatmap (Fig. 17) quantifies the resulting coverage: only ISO 10218-1/2:2025 has non-zero coverage on all six PAIS layers, and its L4a coverage is safety-motivated rather than adversarial; IEC 62443 and the EU AI Act each leave one layer uncovered (L4a and L4b, respectively). Producing AI-specific annexes that treat the Physical AI security stack consistently would need joint work between IEC TC65 (industrial communication networks), ISO TC184/SC2 (robots), and the European Commission's AI Office.

# 9 Open Research Challenges and Future Directions

## 9.1 Per-Layer Coverage

Coverage of the PAIS layers is highly uneven. Of the 620 paper–layer assignments in the corpus (Section 3.5), L3 (Runtime Assurance) and L4b (OT/ICS Security) account for 165 and 202, whereas L4a (HRI) accounts for 57 (Fig. 18, solid bars); the cross-cutting L5 literature is analyzed qualitatively in Section 6.6. Per-layer inter-rater agreement is $\kappa \geq 0.71$ (Section 3.3), and the growth of the L1 sub-topics over 2015–2025 is shown in Fig. 6.

## 9.2 Sensitivity of the Sparsity Findings to the Citation Filter

The eligibility criteria excluded 148 pre-2024 papers with fewer than five citations. This filter is not neutral with respect to the survey's headline finding: recency-poor, citation-poor work is concentrated in nascent areas, which is exactly what L4a (HRI security) and L5 (knowledge feedback) are claimed to be. A filter that suppresses nascent literature and a conclusion that nascent literature is sparse are, without a check, mutually confirming. We therefore bound the effect.

Let $n_\ell$ denote layer assignments in the 312-paper corpus: $n_{\mathrm{L1}} = 85$, $n_{\mathrm{L2}} = 111$, $n_{\mathrm{L3}} = 165$, $n_{\mathrm{L4a}} = 57$, $n_{\mathrm{L4b}} = 202$, over 620 assignments across the five plotted layers. Under *proportional reallocation*, in which the 148 excluded papers carry the observed layer distribution, L4a would gain $148 \times (57/620) \approx 14$ assignments, reaching $\approx$71 of 768, while L4b would gain $\approx$48, reaching $\approx$250. The ordering, and the $> 3\times$ asymmetry between L4a and L4b, are unchanged.

The stronger question is what the filter would have had to hide for the conclusion to fail. For L4a merely to overtake L2, the next-least-covered layer, at least 55 of the 148 excluded papers (37% of everything removed by the citation criterion) would have to be HRI security papers, against an observed L4a base rate of 9.2% in the retained corpus. For L4a to approach L4b, more than 95% of the excluded set would have to be HRI security work. Neither is plausible. We conclude that the L4a and L5 sparsity findings survive the citation filter under any reallocation short of a pathological one.

We state the limits of this argument. It is an analytic bound computed from the retained corpus, not a re-screen of the 148 excluded records, and it therefore cannot detect a systematic difference in kind (as opposed to proportion) between excluded and retained work. The direct test, re-running layer classification with the citation criterion disabled, would be stronger evidence, and the excluded-record list is released with the corpus artifact (Section 11) so that any reader can perform it. Fig. 18 summarizes the comparison.

## 9.3 Research Gap Analysis

Fig. 19 visualizes research coverage versus criticality across the PAIS taxonomy. Underserved areas with high criticality are the research priorities.

## 9.4 Physical AI Security Datasets and Benchmarks

Physical AI security research has no standardized benchmarks that span the PAIS taxonomy. Table 20 reviews the main public datasets and the gaps between them.

The lack of public benchmarks for L1 (hardware trust), L2 (pre-deployment validation), and L4a (HRI security) may be the largest single obstacle to progress. Without them, competing approaches cannot be compared, results cannot be reproduced, and industry has little reason to adopt. The L3/L4b datasets (SWaT, BATADAL, HAI) show what a shared benchmark can do: each has supported dozens of anomaly detection papers. The three gap areas need the same kind of community investment.

*Dataset limitations and scope of empirical claims.* All detection performance figures in Tables 10 and 14 are derived from ICS-specific datasets containing *manually*

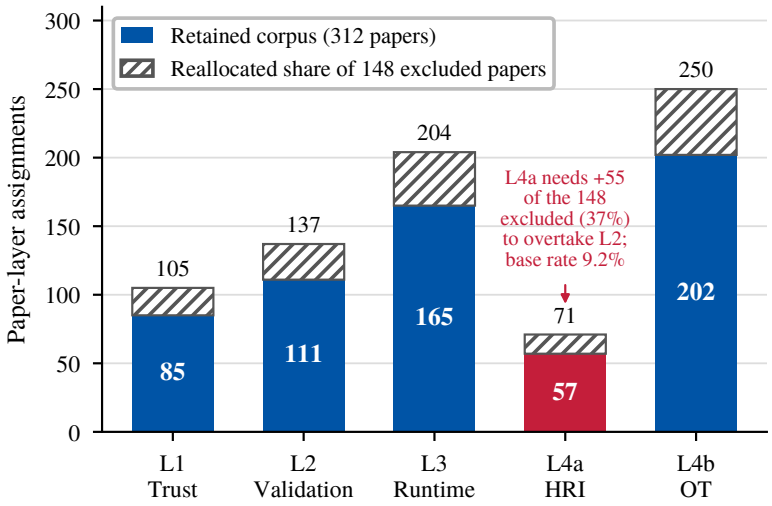



**Fig. 18** Sensitivity of per-layer coverage to the <5-citation, pre-2024 exclusion filter. Solid bars: layer assignments in the retained 312-paper corpus. Hatched extensions: additional assignments under proportional reallocation of all 148 citation-excluded records, with the resulting totals above each bar. The L4a/L4b asymmetry that motivates Research Challenge R4 survives the reallocation.

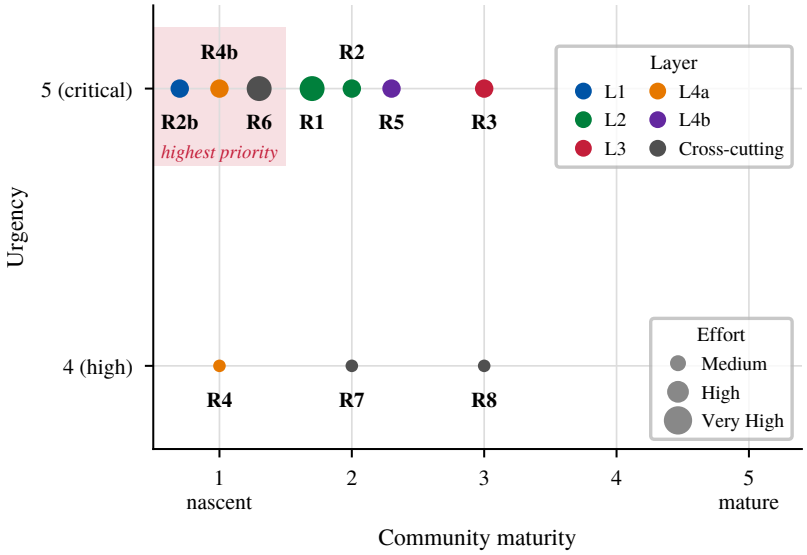



**Fig. 19** Research challenge priorities ($n = 10$: R1–R8, R2b, R4b), plotted from Table 21. Marker size gives estimated effort and color the primary PAIS layer. The shaded cell (maturity 1, urgency 5) is the highest-priority zone. Challenges with identical scores are spread horizontally within their cell for legibility; the underlying ordinal scores are exactly those of Table 21.

**Table 20** Physical AI security datasets and benchmarks: coverage and gaps

| Dataset | Domain | Size | Attack Types Covered | PAIS Layers | Public | Key Limitation |
|---|---|---|---|---|---|---|
| **SWaT (iTrust, 2015) [114]** | Water treatment ICS | 11 days, 51 sensors | Network+sensor attacks | L3, L4b | Yes | No HRI; no L1/L2 attacks |
| **BATADAL (2018) [115]** | Water distribution ICS | 1 year simulated | Actuator manipulation | L3, L4b | Yes | Simulated only; small scale |
| **HAI (KAIST, 2020–22) [116]** | Steam turbine ICS | 3 datasets, 86 tags | Stealthy process attacks | L3, L4b | Yes | OT-only; no AI model layer |
| **ROAD (2022)** | Automotive CAN bus | 3.5h drives | Injection, fuzzing | L3 | Yes | Automotive only; no CPS |
| **HIL-based (custom) [39]** | Robot/cobot cell | Lab-scale | Sensor spoofing, replay | L3, L4a | Rare | Not standardized; few public |
| **OT-Sim datasets [41]** | Generic ICS | Various | Known ICS malware | L3, L4b | Partial | Simulation fidelity gap |
| **GAP: No public dataset** | L1 (trust/attestation) | n/a | Supply chain, firmware | L1 | n/a | No public L1 dataset |
| **GAP: No public dataset** | L2 (validation twin) | n/a | Adversarial twin bypass | L2 | n/a | No public L2/BDT dataset |
| **GAP: No public dataset** | HRI security (L4a) | n/a | Intent spoofing, deepfake | L4a | n/a | No public HRI attack dataset |

We enumerate nine PAIS sub-domains: (1) L1 hardware trust and attestation; (2) L1 supply-chain and firmware provenance; (3) L2 behavioral-twin validation; (4) L2 formal verification of learned controllers; (5) L3 edge anomaly detection; (6) L4a HRI security; (7) L4b OT process-semantic integrity; (8) L4b OT protocol integrity; (9) L5 evidence, invariant-registry, and red-team infrastructure. Only (5) and (7) have a standard public benchmark (SWaT, BATADAL, HAI, ROAD); the remaining *seven of nine* have none. HIL testbeds for L1, L2, and L4a are the highest-priority infrastructure need (R7).

*scripted, rule-based attacks*, not AI-generated adversarial sequences. This has three consequences.

(1) *Attack realism:* SWaT, BATADAL, and HAI attacks were designed by researchers who manually specified anomalous sensor values or commands. Modern AI-crafted attacks (e.g., the DRL-based wear-out sequences of Pasikhani et al. [10]) are statistically indistinguishable from normal operation and are not represented in these corpora. Detection performance on these benchmarks should not be extrapolated to AI-generated threats.
(2) *System diversity:* All three benchmarks cover water treatment or steam turbine process control. Claim transferability to cobots, AMRs, humanoid robots, or food-safety SCADA is untested and should not be assumed.
(3) *HRI gap:* No public benchmark contains HRI-specific attacks (intent spoofing, deepfake commands, coerced-operator scenarios). All performance claims in the HRI section (Section 6.4) are theoretical or extrapolated from adjacent domains.

These limitations do not invalidate the comparative analysis; they set the conditions under which the reported numbers apply. Research Challenge R7 is about closing this gap.

**Table 21** Research challenge priority matrix (R1–R8, R2b, R4b)

| Ref. | Challenge | Layers | Maturity (1–5) | Urgency (1–5) | Effort | Key Open Problem | Enabling Technologies |
|---|---|---|---|---|---|---|---|
| **R1** | Scalable Formal Verification for NNs | L2 | 2 | 5 | Very High (5–10yr) | No tool gives tight certificates beyond $\sim 10^5$ params | Compositional verif., A-G NN reasoning |
| **R2** | Standard Invariant Language (PAIL) | L2, L3, L4 | 2 | 5 | High (3–5yr) | No cross-domain invariant standard | STL/MTL extensions, contract-based design |
| **R2b** | Scalable HW Trojan Detection for AI Chips | L1 | 1 | 5 | High (3–5yr) | No tool scales to SoC-class AI chips | AI-powered side-channel, PUFs, X-ray CT |
| **R3** | Sub-50 ms Edge Anomaly Detection | L3 | 3 | 5 | High (2–4yr) | Accuracy–latency–resource trilemma unsolved | TinyML, NAS for anomaly detectors, HW-SW co-design |
| **R4** | HRI Adversarial Taxonomies & Defenses | L4a | 1 | 4 | Medium (2–3yr) | No formal HRI threat model exists | Multi-modal auth., adv.-robust voice/gesture recognition |
| **R4b** | LLM Security for Physical AI Control | L4a, L2 | 1 | 5 | High (2–4yr) | No Physical AI-specific LLM safety framework | Prompt injection defense, PoisonedRAG mitigation, SSTL |
| **R5** | OT Semantic Integrity at Scale | L4b | 2 | 5 | High (3–5yr) | No deployed general solution | Production-intent spec., trace-level formal verification |
| **R6** | Cross-Layer Security Integration | All | 1 | 5 | Very High (5–10yr) | No peer-reviewed prototype integrates all 5 PAIS layers with empirical validation | Contract-based arch., latency budget analysis |
| **R7** | Physical AI Benchmarks & Datasets | All | 2 | 4 | Medium (2–4yr) | L1, L2, L4a have zero public datasets | HIL testbeds, cobot cells, OT sandboxes |
| **R8** | Regulatory Harmonization | All | 3 | 4 | Medium (2–3yr) | AI Act + IEC 62443 + ISO 10218 not coherently bridged | Joint standards body working groups |

Maturity: 1=nascent, 5=mature. Urgency: 1=low, 5=critical (author assessment based on gap analysis). On these ordinal scales the highest combined urgency-and-effort tier contains exactly R1 and R6 (urgency 5, Very High effort). A second tier of five challenges ties at urgency 5 with High effort: R2, R2b, R3, R4b, and R5. Ties are reported as ties; the ordinal scales do not discriminate within a tier.

## 9.5 Priority Research Agenda

The gap analysis yields ten research challenges (R1–R8, R2b, R4b). Table 21 scores each for maturity, urgency, and estimated effort.

### 9.5.1 R1: Scalable Formal Verification for Neural Network Controllers

Current formal verification techniques give tight, useful safety certificates only for relatively small networks. The limit is *certification tightness*, not a hard parameter ceiling. Relaxation-based tools (CROWN [52], AI$^2$ [24]) can run on large networks with millions of parameters, but their over-approximations become too loose to certify the narrow safety margins required for Physical AI at scale. Exact tools (Reluplex [23, 54], Marabou [53]) provide tight guarantees but face exponential worst-case complexity that makes them intractable beyond $\sim 10^5$ neurons. NNV [22] occupies the middle ground for reachability analysis. The fourth VNN-COMP competition [55] confirms that none of the submitted tools achieves tight certified verification beyond $10^6$ neurons. This *tightness gap* is the central open problem in pre-deployment validation (Fig. 7).

Three directions could narrow the gap: compositional verification (decomposing a large network into independently verifiable sub-networks with assume-guarantee contracts at interfaces); abstraction-refinement techniques that focus exact verification effort on safety-critical network sub-regions; and probabilistic approaches (randomized smoothing [113]) that scale to large models at the cost of statistical rather than formal guarantees. None of these directions has demonstrated tight, formally verified safety certificates for Physical AI-class models under Physical AI latency constraints.

### 9.5.2 R2: Standard Invariant Specification Language for Physical AI (PAIL)

The field lacks a common language for expressing cyber-physical invariants across robotics, OT, and HRI domains. Signal Temporal Logic (STL) [81], Metric Temporal Logic with real-time quantitative bounds [104], and Assume-Guarantee contracts [103] each cover part of the Physical AI invariant space, but none covers all three domains (Table 8). Without a shared invariant language, L2 pre-deployment twins and L3 runtime monitors cannot share invariant definitions across vendor or domain boundaries. A Physical AI Invariant Language (PAIL) should support: real-time temporal properties (MTL/STL-family), spatial constraints (safety envelopes, proximity zones), semantic properties (production intent), and human-authority constraints. Digital twins and runtime monitors should both be able to evaluate it, and its semantics should be standardized so that invariants can be shared across organizations and domains.

PAIL has to trade expressiveness against tractability: a more expressive language allows richer invariants but makes verification and runtime monitoring more expensive. A tiered approach, with a restricted core for real-time runtime checking and a full language for offline verification and twin evaluation, may be necessary.

### 9.5.3 R2b: Scalable Hardware Trojan Detection for AI Chips

AI accelerator chips (GPUs, NPUs) used in Physical AI edge nodes are fabricated at scale in foundries that may not be trusted. No current tool scales hardware Trojan detection to the billions of transistors in a modern AI accelerator SoC. This is

the central open problem at L1, and formal verification, X-ray CT, and power side-channel techniques each address only part of it. Formal verification does not scale to complex modern SoCs; X-ray CT scanning is prohibitively expensive at volume; machine-learning side-channel analysis is near-perfect only for Trojans with distinctive power signatures and falls to roughly 50–77% accuracy for those that blend with normal activity [137]. The Lebanon 2024 incident shows that hardware supply-chain attacks are no longer theoretical. Four research lines have priority:

- AI-powered side-channel power analysis at wafer test [138];
- Hardware fingerprinting using Physical Unclonable Functions (PUFs) for post-deployment identity verification;
- RTL formal verification scaled to subsystem-level AI accelerator components;
- X-ray CT-based post-fabrication verification for high-value Physical AI deployments where cost is justified.

### 9.5.4 R3: Sub-50 ms Anomaly Detection on Resource-Constrained Edges

Achieving high detection accuracy at sub-50 ms latency on embedded compute (Cortex-M, Jetson Nano-class) without cloud connectivity is unsolved. As Table 10 shows, no current technique does well on accuracy, latency, and resource use at once: LSTM/GRU [90, 91] achieves 90–97% accuracy but 20–80 ms latency with 100–500 MB memory; invariant rules achieve <1 ms latency but are blind to novel attacks; autoencoders [92] occupy the middle ground but at higher false-positive rates.

Research directions include neural architecture search (NAS) for embedded anomaly detectors, hardware and software co-design for Physical AI edge nodes, and TinyML approaches [66, 67] that port compressed ML models to microcontroller-class devices. Model compression [68] (pruning, quantization, knowledge distillation) is well studied for classification. Anomaly detection is harder, because the detection threshold has to be re-tuned for the compressed model as well as the architecture. A hybrid, with invariant rules as the primary defense and a lightweight ML detector as the secondary, may be the best practical compromise.

### 9.5.5 R4: HRI Adversarial Attack Taxonomies and Defenses

Intent spoofing, authority escalation, and trust manipulation in HRI have received little study. There is no formal HRI threat model covering the whole attack surface. The corpus has 57 paper–layer assignments for L4a, against 165 for L3 and 202 for L4b, even though HRI security has the largest criticality-to-coverage gap (Fig. 19). Open problems include a formal HRI threat model that covers the full attack surface including coerced operators; authenticated multi-modal intent verification [61]; adversarially robust gesture and voice recognition for robot command interfaces; and deepfake detection methods [64, 65] robust to adversarial synthesis.

Overtrust in autonomous systems [58, 59] is a vulnerability and also a design problem for HRI security: users who overtrust robots do not scrutinize commands, which weakens any intent-verification interface. This makes *transparent design* (architectures

that surface uncertainty and flag high-consequence decisions for human scrutiny) a security principle and not only a usability one [62].

The coerced operator scenario is especially urgent (Table 11): no technical defense exists for the case where a human operator, physically coerced, issues valid commands with genuine biometric liveness. This scenario requires non-technical mitigations (dual-operator consensus for high-consequence commands, anomaly detection on operator access patterns) combined with technical ones.

### 9.5.6 R4b: LLM Security for Physical AI Control

LLMs are entering Physical AI quickly, as robot foundation models, LLM-driven OT planners, and natural-language robot command interfaces. LLM security research has not caught up with the physical consequences. Priority directions are:

- Formal prompt injection detection without LLM internals access (black-box defense) [132];
- Physical consequence-aware jailbreak resistance: safety training that explicitly models physical harm [133];
- PoisonedRAG defense for Physical AI knowledge bases [134];
- OWASP LLM Top 10 mapping to PAIS layers and Physical AI risk tiers;
- Multi-modal injection detection for LLM-integrated HRI (voice + vision + gesture congruence checking).

Synchronous STL (SSTL) [143] could serve as a basis for pre-deployment validation, because the synchronous fragment of STL is decidable. That allows static verification of temporal properties that could not previously be verified. The priorities are SSTL tooling for robotic motion planning, integration of SSTL with behavioral digital twins, and an SSTL property library for common Physical AI safety invariants.

### 9.5.7 R5: OT Semantic Integrity at Scale

No deployed general solution exists for detecting sequences of individually valid commands that compose into harmful outcomes (Fig. 11). Pasikhani et al. [10] demonstrate the problem at scale; Mehmood et al. [20] and the process mining literature [76, 77] provide partial approaches. The gap is the absence of a production-grade semantic integrity monitor that can be deployed in real OT environments with tolerable false-positive rates. Cardenas et al. [36] formalized the theoretical basis for process-aware attack detection, showing that an attacker with process model knowledge can craft attacks that are undetectable by any single-command checker. That result is the reason to analyze semantics at the sequence level.

Research directions include: production-intent specification languages that domain experts can maintain; trace-level formal verification for OT command sequences using STL [107] or process calculi; and DRL-based red-teaming of OT defenses [10] to generate novel wear-out attack variants for testing.

### 9.5.8 R6: Cross-Layer Physical AI Security Integration

The five PAIS layers have been studied separately; no published prototype integrates all five with the contract-based seams (A, B, C) described in Section 7. Integration raises problems that single-layer research does not face: latency budgets across layers (the time budget must partition across L1 attestation, L2 validation, L3 monitoring, and L4 intent checking); contract versioning (when invariants or profiles change, all consuming layers must be consistently updated); and failure mode propagation analysis across layer boundaries. Sandberg et al. [34] discuss the interdependence of security and control-theoretic objectives in networked CPS and point out that optimizing each layer independently can create system-level vulnerabilities at the seams, which is the case for designing the PAIS layers together.

### 9.5.9 R7: Physical AI Security Benchmarks and Datasets

The community lacks standardized benchmarks for Physical AI security evaluation across the full PAIS stack. Existing ICS datasets (SWaT [114], BATADAL [115], HAI [116]) cover L3 anomaly detection but not L1–L2 trust and validation or L4a HRI semantics. New datasets are needed, and they require physical testbeds: collaborative robot cells, OT test environments with adversarial HIL injection, and edge nodes instrumented for hardware trust measurement. The battle-style evaluation methodology of BATADAL [115], which organized a competitive detection challenge with blind evaluation, is a model worth replicating for other PAIS layers. Open HIL testbeds are, in our view, the most useful infrastructure the community could build.

### 9.5.10 R8: Regulatory and Standards Harmonization

Bridging the EU AI Act [86], IEC 62443 [82], ISO 10218 [84], NIST CSF 2.0 [83], and ISO/SAE 21434 [87] into a coherent compliance framework for Physical AI is an open regulatory research problem. The Act's high-risk categorization of Physical AI, combined with its staged high-risk applicability as amended in 2026 (from 2 December 2027 for stand-alone Annex III systems and 2 August 2028 for safety components of products under Article 6(1)) [163], creates compliance obligations that existing standards, written before AI-driven physical control was common, do not meet. Joint working groups between security researchers, standards bodies (IEC TC65, ISO TC184), and regulators are needed to develop AI-specific annexes to existing OT and robotics standards. The compliance heatmap (Fig. 17) shows the size of the gap: no standard fully addresses all six PAIS layers, and L4a (HRI) and L2 (pre-deployment validation) are almost entirely uncovered.

# 10 Cross-Layer Security Implications

## 10.1 The Adversarial Arms Race Between AI Attackers and AI Defenders in Physical AI

Adversaries are increasingly deploying AI against the same defenses that protect Physical AI. The broadest available indicator is cross-domain rather than Physical

AI-specific: CrowdStrike reports an 89% year-on-year rise in attacks by AI-enabled adversaries during 2025 across all domains it observes [148]. Within the Physical AI perimeter itself the evidence is narrower but specific: adaptive malware modifies its own behavior to evade edge anomaly detectors [142]; DRL agents craft attack sequences that outmaneuver statistical defenses [10]; and LLMs generate OT attack code that previously required domain expertise [12]. The cross-domain and Physical AI-specific evidence should not be conflated: the former establishes a trend in adversary tooling, the latter establishes exploitability of Physical AI defenses in particular.

For defenders, static rule-based defenses will not be enough. Physical AI security needs *continuously adaptive* defenses that learn from operational telemetry (L3), adversarial red-team scenarios (L5), and community threat intelligence (MITRE ATLAS contributions). This is why the L5 knowledge-feedback layer is part of the architecture and not an optional extra: it is how the PAIS stack can adapt at least as fast as the attackers do.

### 10.2 The Case for a Federated Architecture

Physical AI deployments carry data that operators will not pool: commercially sensitive production intent, workforce-safety records, proprietary robot trajectories. At fleet scale this rules out centralized security monitoring. Federated learning provides a way to collaborate on security without centralizing the data.

Federated intrusion detection is established for industrial settings [15, 42, 165]; extending federation to pre-deployment validation and red-teaming is a natural next step that has not yet been demonstrated for Physical AI. A federated PAIS, where L3 anomaly models, L2 invariant registries, and L5 evidence stores are collaboratively maintained across fleet members without sharing raw OT telemetry, may well be the only workable route to fleet-wide Physical AI security for large industrial operators. Tamper-evident evidence can be added cheaply: Mishra et al. report that SHA-256 hash-chained trust logging adds negligible latency to an IIoT intrusion detector [29], but their design is centralized, so a federated evidence layer remains an open problem.

### 10.3 Defense-in-Depth Across PAIS Layers

Table 22 summarizes the defense-in-depth properties of the PAIS taxonomy: the security contribution of each layer when other layers fail, and the residual risk that remains even when a layer functions correctly. No single layer gives complete protection, and each layer's residual is covered, at least in part, by another. That is the argument for the five-layer design; how much stronger the combination is in practice has yet to be measured (experiment E3).

## 11 Conclusion

This paper began from the observation that research on Physical AI security is fragmented: hardware trust, AI assurance, runtime monitoring, human–robot interaction, OT semantics, and security knowledge management are each studied in depth, but

**Table 22** PAIS defense-in-depth: security contribution and failure residual per layer

| Layer | Security contribution | Failure residual |
|---|---|---|
| L1 | Blocks unsigned/tampered decisions at source | Attested-but-poisoned model reaches L2 |
| L2 | Blocks physically unsafe decisions before actuation | Safe-seeming but semantically wrong actions reach L3 |
| L3 | Detects runtime deviations; triggers containment | Stealthy deviations below threshold escape |
| L4a | Validates human intent; resists HRI manipulation | Sophisticated deepfakes bypass single-modality checks |
| L4b | Detects OT semantic intent violations | Novel attack compositions not in invariant model escape |
| L5 | Captures evidence; improves invariants; red-teams | Stale knowledge; poisoned evidence store degrades all layers |

by different communities and without a shared account of how they depend on one another. The PAIS taxonomy addresses that fragmentation by giving each dimension a defined security role, an explicit boundary, and stated interfaces to the others. As a classification frame, it made a scattered literature comparable: the same five layers organize the 312-paper corpus, the threat classification, the MITRE, framework, standards, and regulatory mapping, and the reference architecture, so that a gap observed in one view can be checked against the others. Several of these dimensions have been surveyed individually, and in places more deeply, by the works in Table 1; the contribution here is the structure that connects them.

The systematic evidence shows that maturity is uneven in a specific way. The layers most thoroughly studied, runtime assurance and OT semantics (165 and 202 paper–layer assignments), are those inherited from classical CPS and ICS security, and even there performance evidence rests on three process-control benchmarks with hand-scripted attacks. The layers where Physical AI differs most from its predecessors are the least mature on every measure the review applied. Model provenance and the validation of learned controllers remain open at L1 and L2. HRI security accounts for 57 assignments, and every L4a and L5 threat type rests on analogous or theorized evidence, as do 14 of the 19 threat types overall (Table 5). Seven of nine PAIS subdomains have no public benchmark, and no knowledge base, framework, or standard addresses L4a beyond safety-motivated coverage that does not model an adversary. No framework or published prototype integrates all five layers, and the latency analysis of Section 7.3 indicates that integration is a binding engineering constraint rather than a formality.

The research agenda is shaped by these findings, and its challenges are interdependent. Benchmarks and testbeds (R7) are the precondition for measurable progress on every other challenge. Scalable verification (R1) and a shared invariant language (R2) determine whether validation can keep pace with learned controllers, and R2 in turn supplies the specification infrastructure that HRI security (R4, R4b) and OT semantic integrity (R5) both need. Hardware Trojan detection (R2b) and edge anomaly

detection within control-loop budgets (R3) address the platform and runtime conditions under which those checks must operate. Cross-layer integration (R6) can be evaluated only once these components mature, and regulatory harmonization (R8) depends on the evidence that R1–R7 produce. Progress along this sequence is what would allow Physical AI systems to be *secure* in the sense that each decision's origin can be verified, *trustworthy* in the sense that its safety and intent are validated before actuation, and *resilient* in the sense that deviations are contained locally and turned into knowledge that strengthens the system rather than degrading it.

A survey can propose a structure and draw a first map. Whether the PAIS framing lasts will depend on empirical progress on these challenges, starting with the benchmarks and testbeds that would let the claims in this paper be tested.

## Author Contributions and AI disclosure

*Author contributions.* A1 designed the search protocol and performed independent title/abstract screening, full-text eligibility assessment, and first-pass layer classification. A2 and A3 adjudicated classification disagreements and did not participate in first-pass coding. A1 drafted the taxonomy and Layers 1–3; A2 drafted Layers 4a, 4b, and 5 and the standards analysis; A1, A2 and A3 contributed the OT and hardware-trust material and reviewed all quantitative claims against their sources. All authors approved the final manuscript. GPT-4o was used for copyediting assistance only. All technical content, proofs, experimental designs, and reported results are the authors' own.

## Acknowledgements

We thank Gregory Epiphaniou and Dinh Truong, both at the University of Warwick, for their constructive comments and suggestions, which helped improve the clarity and presentation of this paper.

## Limitations

This survey has two acknowledged limitations. (1) *Rapidly evolving threat landscape*: AI-specific attack tools described as "emerging" may have been operationalized by the time of publication. (2) *PRISMA compliance*: this review follows PRISMA 2020 reporting guidelines for search documentation and exclusion flow but does not achieve full PRISMA 2020 protocol compliance (no prospective registration, no formal data extraction form, no within-study risk-of-bias assessment). It should be characterized as a "PRISMA-informed structured literature review."

## Declarations

### Funding

Funding: Not applicable.

## Competing interests

The authors declare that they have no competing interests.

# References


[1] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," in *Proc. 3rd USENIX Workshop on Hot Topics in Security (HotSec)*, 2008.

[2] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[3] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Proc. 1st Workshop on Secure Control Systems (SCS)*, 2010.

[4] A. Haskard and D. C. Herath, "Secure robotics: navigating challenges at the nexus of safety, trust, and cybersecurity in cyber-physical systems," *ACM Computing Surveys*, vol. 57, no. 9, 2025.

[5] S. Neupane et al., "Security in AI-Robotics: A comprehensive survey," arXiv preprint arXiv:2310.08565, 2023.

[6] Industrial Cyber, "Rising ICS incidents drive shift from reactive risk models to intelligence-driven OT security strategies," Mar. 2026. [Online]. Available: https://industrialcyber.co

[7] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, "Adversarial attacks on machine learning cybersecurity defences in industrial control systems," *Journal of Information Security and Applications*, vol. 58, 2021.

[8] G. Zizzo, C. Hankin, S. Maffeis, and K. Jones, "Adversarial attacks on time-series intrusion detection for industrial control systems," in *Proc. IEEE Int. Conf. on Trust, Security and Privacy*, 2020.

[9] A. Botta, S. Rotbei, S. Zinno, and G. Ventre, "Cyber security of robots: a comprehensive survey," *Intelligent Systems with Applications*, vol. 18, 2023.

[10] A. Pasikhani, P. Gope, Y. Yang, S. Mehnaz, and B. Sikdar, "Baiting AI: Deceptive adversary against AI-protected industrial infrastructures," arXiv preprint arXiv:2601.08481, Jan. 2026.

[11] K. Eykholt et al., "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE CVPR*, 2018, pp. 1625–1634.

[12] G. Deng et al., "PentestGPT: Evaluating and harnessing large language models for automated penetration testing," in *Proc. 33rd USENIX Security Symp.*, 2024, pp. 847–864.

[13] J. Ménétrey et al., "An exploratory study of attestation mechanisms for trusted execution environments," in *Proc. SysTEX Workshop*, 2022, arXiv:2204.06790.

[14] M. Schneider et al., "SoK: Hardware-supported trusted execution environments," arXiv:2205.12742, 2022.

[15] V. Mothukuri et al., "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2022.

[16] X. Chen, W. Feng, N. Ge, and Y. Zhang, "Zero trust architecture for 6G security," *IEEE Network*, 2024.

[17] M. Eckhart, A. Ekelhart, and E. Weippl, "Enhancing cyber situational awareness for cyber-physical systems through digital twins," in *Proc. IEEE ETFA*, 2019.

[18] A. De Benedictis, C. Esposito, and A. Somma, "Toward the adoption of secure cyber digital twins to enhance cyber-physical systems security," in *Proc. QUATIC 2022*, Springer, 2022.

[19] D. Cho, J. Cho, and W. Kim, "A generative digital twin for continually enhancing the intended functional safety of cyber-physical systems," *IEEE Trans. Reliability*, vol. 74, no. 2, pp. 2561–2575, Jun. 2025.

[20] M. Mehmood, Z. Baig, and N. Syed, "Securing industrial control systems (ICS) through attack modelling and rule-based learning," in *Proc. COMSNETS 2024*.

[21] M. A. Al-Garadi et al., "A digital twin-based approach for detecting cyber attacks in ICS," *Applied Sciences*, vol. 14, no. 19, 2024.

[22] H.-D. Tran et al., "NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems," in *Proc. CAV 2020*.

[23] G. Katz et al., "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. CAV 2017*.

[24] T. Gehr et al., "AI$^2$: Safety and robustness certification of neural networks with abstract interpretation," in *Proc. IEEE S&P*, 2018.

[25] P. Kuruppuarachchi et al., "Trust-based digital twin behavioural categorisation in a collaborative ecosystem," in *Proc. IEEE CloudCom 2025*.

[26] X. Jia et al., "Deep anomaly detection for time series," *Computer Science Review*, vol. 58, 2025.

[27] X. Zhao et al., "Securing AI agents in cyber-physical systems: A survey of environmental interactions, deepfake threats, and defenses," arXiv preprint arXiv:2601.20184, Jan. 2026. [Online]. Available: https://arxiv.org/abs/2601.20184

[28] Z. E. Huma et al., "Adversarial machine learning in IoT security: A comprehensive survey," *ACM Computing Surveys*, 2025.

[29] S. Mishra, T. S. M. Aldafas, and N. S. Alshammari, "A zero-trust digital twin framework for privacy-preserving multi-dataset intrusion detection in industrial IoT with lightweight blockchain auditing," *Scientific Reports*, vol. 16, Art. 15236, 2026, doi:10.1038/s41598-026-42041-w.

[30] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[31] J. Giraldo et al., "A survey of physics-based attack detection in cyber-physical systems," *ACM Computing Surveys*, vol. 51, no. 4, pp. 76:1–76:36, 2018.

[32] S. McLaughlin et al., "The cybersecurity landscape in industrial control systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1039–1057, 2016.

[33] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security: a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[34] H. Sandberg, S. Amin, and K. H. Johansson, "Cyberphysical security in networked control systems: an introduction to the issue," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 20–23, 2015.

[35] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[36] A. A. Cárdenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proc. 6th ACM Symp. Information, Computer and Communications Security (ASIACCS)*, pp. 355–366, 2011.

[37] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, "Guide to industrial control systems (ICS) security," NIST Special Publication 800-82 Rev. 3, Sep. 2023. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-82r3

[38] E. Byres and J. Lowe, "The myths and facts behind cyber security risks for industrial control systems," in *Proc. ISA VDE Congress*, 2004.

[39] H.-Y. Shin, S. Lee, J.-H. Yun, and H. Kim, "Two ICS security datasets and a comparison of anomaly detection methods," in *Proc. 11th Int. Conf. ICT Systems Security and Privacy Protection (SEC)*, 2020.

[40] M. Kravchik and A. Shabtai, "Efficient cyber attack detection in industrial control systems using lightweight neural networks and PCA," *IEEE Trans. Dependable*

*and Secure Computing*, vol. 18, no. 2, pp. 524–536, 2021.

[41] S. Adepu and A. Mathur, "An investigation into the response of a water treatment system to cyber attacks," in *Proc. 17th IEEE Int. Symp. High Assurance Systems Engineering (HASE)*, 2016.

[42] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Trans. Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[43] P. Nader, P. Honeine, and P. Beauseroy, "$\ell_p$-norms in one-class classification for intrusion detection in SCADA systems," *IEEE Trans. Industrial Informatics*, vol. 10, no. 4, pp. 2308–2317, 2014.

[44] I. N. Fovino, A. Carcano, M. Masera, and A. Trombetta, "An experimental investigation of malware attacks on SCADA systems," *Int. J. Critical Infrastructure Protection*, vol. 2, no. 4, pp. 139–145, 2009.

[45] S. Nazir, S. Patel, and D. Patel, "Assessing and augmenting SCADA cyber security: A survey of techniques," *Computers & Security*, vol. 70, pp. 436–454, 2017.

[46] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. 6th Int. Conf. Learning Representations (ICLR)*, 2018.

[47] Trusted Computing Group, "TPM library specification, family 2.0, level 00, revision 01.59," TCG, Sep. 2019. [Online]. Available: https://trustedcomputinggroup.org

[48] OpenTitan Contributors, "OpenTitan: An open-source silicon root of trust," lowRISC, 2023. [Online]. Available: https://opentitan.org

[49] J. Harrison, S. Lerner, J. Bier, and A. Sherwood, "Principled security hardening of OpenTitan," in *Proc. IEEE Secure Development Conf. (SecDev)*, 2020.

[50] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel," in *Proc. 28th USENIX Security Symp.*, pp. 515–532, 2019.

[51] A. Dubey, R. Cammarota, and A. Aysu, "MaskedNet: The first hardware inference engine aiming power side-channel protection," in *Proc. IEEE Int. Symp. Hardware Oriented Security and Trust (HOST)*, pp. 197–208, 2020.

[52] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[53] G. Katz et al., "The Marabou framework for verification and analysis of deep neural networks," in *Proc. 31st Int. Conf. Computer Aided Verification (CAV)*, Springer, LNCS 11561, pp. 443–452, 2019.

[54] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: a calculus for reasoning about deep neural networks," *Formal Methods in System Design*, vol. 60, pp. 87–116, 2022.

[55] C. Brix et al., "The fourth international verification of neural networks competition (VNN-COMP 2023)," arXiv:2312.16760, 2023.

[56] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technology, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.

[57] M. Eckhart and A. Ekelhart, "A specification-based state replication approach for digital twins," in *Proc. ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*, pp. 36–47, 2018.

[58] P. Robinette, W. Li, R. Allen, A. M. Howard, and A. R. Wagner, "Overtrust of robots in emergency evacuation scenarios," in *Proc. 11th ACM/IEEE Int. Conf. Human-Robot Interaction (HRI)*, pp. 101–108, 2016.

[59] P. A. Hancock et al., "A meta-analysis of factors affecting trust in human-robot interaction," *Human Factors*, vol. 53, no. 5, pp. 517–527, 2011.

[60] E. Short, J. Hart, M. Vu, and B. Scassellati, "No fair!! An interaction with a cheating robot," in *Proc. 5th ACM/IEEE Int. Conf. Human-Robot Interaction (HRI)*, pp. 219–226, 2010.

[61] S. Nikolaidis, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in collaborative tasks: Models and experiments," *Int. J. Robotics Research*, vol. 36, no. 5–7, pp. 618–634, 2017.

[62] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[63] J. Urakami and K. Seaborn, "Nonverbal cues in human-robot interaction: A communication studies perspective," *ACM Trans. Human-Robot Interaction*, vol. 12, no. 2, Art. 22, 2023.

[64] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake detection: A systematic literature review," *IEEE Access*, vol. 10, pp. 25494–25513, 2022.

[65] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[66] J. Lin et al., "MCUNet: Tiny deep learning on IoT devices," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[67] C. Banbury et al., "MLPerf Tiny benchmark," in *Proc. 35th Conf. Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[68] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. 4th Int. Conf. Learning Representations (ICLR)*, 2016.

[69] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST Special Publication 800-207, 2020.

[70] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, "Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust," *Computers & Security*, vol. 110, 2021.

[71] J. Wang, K. T. Cheng, and M. M. T. Chakrabarti, "Supply chain attacks on hardware: A review," *Computer*, vol. 54, no. 6, pp. 28–37, 2021.

[72] J.-C. Hu, S.-H. Lee, R. Hou, and M.-H. Chen, "A survey on blockchain for information systems management and security," *Information Processing & Management*, vol. 58, no. 1, 2021.

[73] M. Ali and P. Mahrawy, "Blockchain for cybersecurity and privacy: Architectures, challenges, and future directions," *IEEE Access*, vol. 7, pp. 136217–136243, 2019.

[74] G. Rathee et al., "A secure communicating things network framework for industrial IoT using blockchain technology," *IEEE Access*, vol. 9, pp. 69–80, 2021.

[75] R. Tian et al., "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proc. 40th Int. Conf. Software Engineering (ICSE)*, pp. 303–314, 2018.

[76] W. M. P. van der Aalst, "Process mining: Discovery, conformance and enhancement of business processes," Springer, 2011.

[77] A. Das, M. Adepu, and J. Zhou, "A process-mining approach for detecting anomalies in the SWaT dataset," *IEEE Trans. Information Forensics and Security*, vol. 17, pp. 3544–3557, 2022.

[78] A. Donzé, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in *Proc. 22nd Int. Conf. Computer Aided Verification (CAV)*, Springer, LNCS 6174, pp. 167–170, 2010.

[79] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-TaLiRo: A tool for temporal logic falsification for hybrid systems," in *Proc. 17th Int. Conf.*

*Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Springer, LNCS 6605, pp. 254–257, 2011.

[80] K. G. Larsen, P. Pettersson, and W. Yi, "UPPAAL in a nutshell," *Int. J. Software Tools for Technology Transfer*, vol. 1, no. 1–2, pp. 134–152, 1997.

[81] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Proc. Int. Conf. Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems (FORMATS/FTRTFT)*, Springer, LNCS 3253, pp. 152–166, 2004.

[82] IEC, "Industrial automation and control systems security (IEC 62443 series)," International Electrotechnical Commission, 2018–2020.

[83] NIST, "Cybersecurity framework 2.0," National Institute of Standards and Technology, Feb. 2024.

[84] ISO/TC 184/SC 2, "Robots and robotic devices: safety requirements for industrial robots, Part 1: Robots (ISO 10218-1:2011)," International Organization for Standardization, 2011.

[85] ISO/TC 184/SC 2, "Robots and robotic devices: collaborative robots (ISO/TS 15066:2016)," International Organization for Standardization, 2016.

[86] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Official Journal of the European Union*, L, 2024.

[87] ISO/SAE, "Road vehicles: cybersecurity engineering (ISO/SAE 21434:2021)," International Organization for Standardization, 2021.

[88] NIST, "Security and privacy controls for information systems and organizations (NIST SP 800-53 Rev. 5)," National Institute of Standards and Technology, Sep. 2020.

[89] ISO, "Safety of machinery, safety-related parts of control systems, part 1: General principles for design (ISO 13849-1:2023)," International Organization for Standardization, 2023.

[90] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[91] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[92] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. Network and Distributed System Security Symp. (NDSS)*, 2018.

[93] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in IoT systems," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10327–10335, 2021.

[94] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection systems," in *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD)*, Springer, 2022.

[95] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

[96] K. Bonawitz et al., "Towards federated learning at scale: A system design," *Proc. Machine Learning and Systems*, 2019.

[97] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, pp. 634–643, 2019.

[98] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. 38th IEEE Symp. Security and Privacy (S&P)*, pp. 3–18, 2017.

[99] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models through prediction APIs," in *Proc. 25th USENIX Security Symp.*, pp. 601–618, 2016.

[100] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Computer and Communications Security (CCS)*, pp. 1322–1333, 2015.

[101] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proc. 26th ACM SIGSAC Conf. Computer and Communications Security (CCS)*, pp. 241–257, 2019.

[102] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[103] S. A. Seshia et al., "Formal specification for deep neural networks," in *Proc. 16th Int. Symp. Automated Technology for Verification and Analysis (ATVA)*, Springer, LNCS 11138, pp. 20–34, 2018.

[104] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Nickovic, and S. Sankaranarayanan, "Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications," in *Lectures on Runtime Verification*, Springer, LNCS 10457, pp. 135–175, 2018.

[105] D. Tran, H.-D. Tran, D. Lopez, H. Nguyen, L. Everett, and T. T. Johnson, "Verification of piecewise deep neural networks: a star set approach with zonotope pre-filter," *Formal Aspects of Computing*, vol. 33, no. 4–5, 2021.

[106] X. Sun et al., "Temporal logic for runtime verification of safety monitors in autonomous systems," in *Proc. 19th Int. Conf. Formal Aspects of Component Software (FACS)*, 2022.

[107] A. Donzé, T. Ferrère, and O. Maler, "Efficient robust monitoring for STL," in *Proc. 25th Int. Conf. Computer Aided Verification (CAV)*, Springer, LNCS 8044, pp. 264–279, 2013.

[108] T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[109] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2023.

[110] A. Happe and J. Cito, "Getting pwned by AI: Penetration testing with large language models," in *Proc. 31st ACM Joint Meeting on European Software Engineering Conf. and Symp. on Foundations of Software Engineering (FSE)*, pp. 2082–2086, 2023.

[111] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," arXiv:2211.09527, 2022.

[112] I. Evtimov et al., "Robust physical-world attacks on machine learning models," arXiv:1707.08945, 2017.

[113] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness through randomized smoothing," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, pp. 1310–1320, 2019.

[114] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. Int. Workshop Cyber-physical Systems for Smart Water Networks (CySWater)*, pp. 31–36, 2016.

[115] R. Taormina et al., "The battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks," *J. Water Resources Planning and Management*, vol. 144, no. 8, 2018.

[116] H.-Y. Shin, S. Lee, J.-H. Yun, and H. Kim, "HAI 1.0: HIL-based augmented ICS security dataset," in *Proc. 13th USENIX Workshop on Cyber Security*

*Experimentation and Test (CSET)*, 2020.

[117] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[118] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. 1st IEEE European Symp. Security and Privacy (EuroS&P)*, pp. 372–387, 2016.

[119] Dragos, Inc., "Dragos 2025 OT/ICS Cybersecurity Report: 8th Annual Year in Review," Dragos, Inc., Feb. 2025. [Online]. Available: https://www.dragos.com/year-in-review (1,693 ransomware attacks on industrial organizations in 2024, +87% year on year; ransomware groups targeting OT 50 → 80.)

[120] Cybersecurity and Infrastructure Security Agency (CISA), "2024 CISA cybersecurity advisory and threat assessment," U.S. Department of Homeland Security, 2024. [Online]. Available: https://www.cisa.gov

[121] ABI Research, "Robotics market outlook: USD 50 billion (2025) to USD 110.7 billion (2030), 13.8% CAGR," ABI Research market data, 2025. *Vendor market estimate.*

[122] MarketsandMarkets, "Operational technology (OT) security market: global forecast to 2031: USD 27.39 billion (2026) to USD 58.94 billion (2031)," MarketsandMarkets, 2026. *Vendor market estimate.*

[123] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, n71, 2021.

[124] MITRE Corporation, "MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems): `atlas-data` releases (v5.1.0, Nov. 2025, and later) and CHANGELOG," 2025–2026. [Online]. Available: https://atlas.mitre.org and https://github.com/mitre-atlas/atlas-data

[125] B. E. Strom et al., "MITRE ATT&CK: Design and philosophy," MITRE Corporation Technical Report, 2020.

[126] MITRE Corporation, "D3FEND: A knowledge graph of cybersecurity countermeasures," v1.3.0, Dec. 2025. [Online]. Available: https://d3fend.mitre.org (267 defensive techniques across seven tactics; includes an operational technology extension.)

[127] MITRE Corporation, Niyo Little Thunder Pearson, Red Balloon Security, and Narf Industries, "EMB3D threat model for embedded devices," May 2024. [Online]. Available: https://emb3d.mitre.org

[128] MITRE Corporation, "CAPEC: Common Attack Pattern Enumeration and Classification," [Online]. Available: https://capec.mitre.org

[129] MITRE Corporation, "CWE: Common Weakness Enumeration," [Online]. Available: https://cwe.mitre.org

[130] H. Srinivasan and M. Karimi, "Threat-based security controls to protect industrial control systems using MITRE ATT&CK for ICS framework," *IEEE Access*, vol. 13, 2025.

[131] W. Zhang et al., "A study on prompt injection attack against LLM-integrated mobile robotic systems," arXiv:2408.03515, 2024.

[132] T. Geng, Y. Qu, and W. E. Wong, "A white-box prompt injection attack on embodied AI agents driven by large language models," *Journal of Systems and Software*, vol. 235, Art. 112782, May 2026, doi:10.1016/j.jss.2026.112782.

[133] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications (LLM01:2025 Prompt Injection)," 2025. [Online]. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications

[134] W. Zou, R. Guo, B. Liu, J. Yuan, M. Zhang, and C. Gao, "PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models," arXiv:2402.07867, 2024.

[135] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," arXiv:2307.15043, 2023.

[136] A. Chatterjee, "Hardware Trojans in the physical era: lessons from the Lebanon pager incident," *IET Cyber-Physical Systems: Theory & Applications*, 2025.

[137] S.-N. Puspa, A. Enan, R. Majumdar, M. S. Salek, G. Comert, and M. Chowdhury, "An AI-enabled side channel power analysis based hardware trojan detection method for securing the integrated circuits in cyber-physical systems," arXiv:2411.12721, 2024.

[138] IDS Technologies, "Silicon fingerprinting and AI-powered wafer-test anomaly detection," Technical Brief, 2025.

[139] CISA and DoD, "Zero Trust for OT activities and outcomes: DoD Zero Trust Maturity Model extension for ICS/SCADA," 2025. [Online]. Available: https://www.cisa.gov/zero-trust-maturity-model

[140] Zentera Systems, "Zero trust microsegmentation for legacy OT environments," 2025. [Online]. Available: https://www.zentera.net

[141] ColorTokens, "Progressive zero trust deployment for operational technology," 2025. [Online]. Available: https://www.colortokens.com

[142] ShieldWorks Cyber, "Adaptive AI-driven malware in OT environments: 2026 industrial threat intelligence report," ShieldWorks Cyber, Feb. 2026.

[143] D. Neider and G. Weissenbacher, "Synthesizing and verifying temporal properties of reactive systems using synchronous signal temporal logic," *Formal Methods in System Design*, Springer, 2024.

[144] S. Jakšić et al., "Quantitative monitoring of STL with edit distance," in *Proc. 18th Int. Conf. Runtime Verification (RV)*, Springer, 2018.

[145] L. Lindemann, G. J. Pappas, and D. V. Dimarogonas, "Reactive and risk-aware control for signal temporal logic," *IEEE Trans. Automatic Control*, vol. 68, no. 6, pp. 3322–3337, 2023.

[146] T. Visconti et al., "Robust monitoring of STL specifications under sensor uncertainty," *Sensors*, vol. 22, no. 14, 2022.

[147] Dragos, Inc., "Dragos 2026 OT/ICS Cybersecurity Report and Year in Review," Dragos, Inc., Feb. 2026. [Online]. Available: https://www.dragos.com/year-in-review (3,300 industrial organizations affected by ransomware in 2025; 119 tracked ransomware groups.)

[148] CrowdStrike, "CrowdStrike 2026 Global Threat Report," CrowdStrike Holdings, Feb. 2026. [Online]. Available: https://www.crowdstrike.com/global-threat-report/ (89% year-on-year increase in attacks by AI-enabled adversaries, all domains.)

[149] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. 16th ACM Workshop on Artificial Intelligence and Security (AISec)*, pp. 79–90, 2023.

[150] A. Brohan et al., "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Proc. 7th Conf. on Robot Learning (CoRL)*, 2023.

[151] D. Driess et al., "PaLM-E: An embodied multimodal language model," in *Proc. 40th Int. Conf. on Machine Learning (ICML)*, 2023.

[152] M. Ahn et al., "Do as I can, not as I say: Grounding language in robotic affordances," in *Proc. 6th Conf. on Robot Learning (CoRL)*, 2022.

[153] ISO/TC 184/SC 2, "Robotics, safety requirements, Part 1: Industrial robots (ISO 10218-1:2025)," International Organization for Standardization, Feb. 2025.

[154] ISO/TC 184/SC 2, "Robotics, safety requirements, Part 2: Industrial robot applications and robot cells (ISO 10218-2:2025)," International Organization for Standardization, Feb. 2025. (Incorporates the collaborative-application content of ISO/TS 15066:2016; biomechanical limits appear as informative Annex M.)

[155] W. Ozga, P. Sagmeister, T. Visegrády, and S. Dragone, "Scalable attestation of virtualized execution environments in hybrid- and multi-cloud," arXiv:2304.00382, 2023 (published as "Wawel," in *Proc. IEEE CLOUD*, pp. 96–107, 2023).

[156] H. Raj et al., "fTPM: A software-only implementation of a TPM chip," in *Proc. 25th USENIX Security Symp.*, pp. 841–856, 2016.

[157] D. Moghimi, B. Sunar, T. Eisenbarth, and N. Heninger, "TPM-FAIL: TPM meets timing and lattice attacks," in *Proc. 29th USENIX Security Symp.*, pp. 2057–2073, 2020.

[158] S. Wang et al., "Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 29909–29921, 2021.

[159] S. Huang et al., "Adversarial attacks on neural network policies," in *Proc. ICLR Workshop on Adversarial Examples*, 2017.

[160] Y. Cao et al., "Adversarial sensor attack on LiDAR-based perception in autonomous driving," in *Proc. 26th ACM Conf. Computer and Communications Security (CCS)*, pp. 2267–2281, 2019.

[161] Oxmaint, "Industrial cybersecurity: OT, ICS & SCADA security guide," Apr. 2026. [Online]. Available: https://oxmaint.com

[162] Q. Xu, S. Ali, and T. Yue, "Digital twin-based anomaly detection with curriculum learning in cyber-physical systems," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 5, Article 113, Jul. 2023.

[163] European Parliament and Council of the European Union, "Regulation (EU) 2026/1744 amending Regulation (EU) 2024/1689 as regards the simplification of the implementation of harmonised rules on artificial intelligence (Digital Omnibus on AI)," *Official Journal of the European Union*, L, 24 Jul. 2026.

[164] WUSF Public Media, "Official says 'hack' of Oldsmar city water treatment plant in 2021 didn't happen," Apr. 2023, reporting an FBI statement that the investigation could not confirm a targeted cyber intrusion. [Online]. Available: https://www.wusf.org/local-state/2023-04-13/official-hack-oldsmar-city-water-treatment-plant-2021

[165] P. Ruzafa-Alcázar, P. Fernández-Saura, E. Mármol-Campos, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal-Bernabe, and A. F. Skarmeta, "Intrusion detection based on privacy-preserving federated learning for the industrial IoT," *IEEE Trans. Industrial Informatics*, vol. 19, no. 2, pp. 1145–1154, 2023, doi:10.1109/TII.2021.3126728.