



ARTICLE

A Digital Twin and Lightweight Blockchain Logging Framework for Real-Time CPS Threat Detection

Salam Al-E'mari^{1,*}, Yousef Sanjalawe^{2,*}, Budoor Allehyani³, Ahd Aljarf⁴ and Fares Alharbi⁵

¹Department of Information Security, Faculty of Information Technology, University of Petra (UoP), Amman, Jordan

²Department of Information Technology, King Abdullah II School for Information Technology, University of Jordan (JU), Amman, Jordan

³Department of Software Engineering, College of Computing, Umm Al-Qura University (UQU), Makkah, Saudi Arabia

⁴Department of Cybersecurity, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia

⁵Department of Computer Science, College of Computing and IT, Shaqra University, Shaqra, Saudi Arabia

*Corresponding Authors: Salam Al-E'mari. Email: salam.ammari@uop.edu.jo; Yousef Sanjalawe. Email: y.sanjalawe@ju.edu.jo

Received: 01 June 2026; Accepted: 24 August 2026; Published: 22 October 2026

ABSTRACT: Cyber-Physical Systems (CPS), including Industrial Internet of Things (IIoT) environments and smart infrastructures, are increasingly vulnerable to sophisticated cyberattacks due to their heterogeneous architectures, large-scale deployments, and strict real-time operational constraints. Conventional security mechanisms often struggle to provide low-latency threat detection, continuous monitoring, and reliable forensic traceability under resource-constrained conditions. This paper presents a lightweight Digital Twin (DT)-driven cybersecurity framework that integrates synchronized DT state modeling, hybrid anomaly detection, and tamper-evident audit logging for real-time CPS threat detection. Within this framework, the DT is realized as an operational, data-driven twin whose role is continuous bidirectional state synchronization, integrity-aware feature abstraction, and context-aware representation of CPS assets for detection, rather than a physics-based simulation, “what-if” scenario, or predictive-maintenance twin; these richer capabilities are enabled by the modular DT design and are identified as future extensions. The proposed framework employs a decoupled training-deployment strategy in which Isolation Forest (IF), Gradient Boosting Machine (GBM), and Multi-Layer Perceptron (MLP) models are combined during an offline ensemble calibration stage to optimize detection thresholds, while only a lightweight MLP model is executed during runtime inference to minimize computational overhead. To enhance detection reliability and interpretability, the learning-based detection pipeline is further augmented with domain-aware rule-based verification. In addition, validated security events are recorded via a lightweight, hash-chained, append-only logging mechanism that provides tamper-evident forensic traceability without the latency overhead of consensus-driven distributed ledger systems. Experimental evaluation on controlled synthetic CPS telemetry achieved 99.6% detection accuracy, 99.5% precision, 99.8% recall, and an F1-score of 99.6%. The framework was further validated on three real-world industrial CPS benchmarks (SWaT, WADI, and BATADAL), where F1-scores ranged from 0.976 to 0.986, demonstrating consistent generalization across heterogeneous operational environments. In addition, the proposed architecture sustained telemetry processing rates exceeding 5000 events per minute with an average logging latency of 0.84 s and CPU utilization below 65%. These results demonstrate that the proposed framework provides an efficient and practically deployable CPS security architecture that balances lightweight runtime inference, detection reliability, and forensic auditability in resource-constrained environments.

KEYWORDS: Blockchain; cyber-physical systems; digital twins; hybrid anomaly detection; industrial Internet of Things

1 Introduction

Cyber-physical systems (CPS) have evolved from simple sensor-actuator networks into large-scale, tele-monitored infrastructures—spanning smart grids, water treatment plants, and city-scale Industrial Internet of Things (IIoT) deployments—whose heterogeneity, tight real-time constraints, and safety-critical operations introduce a fundamentally different threat surface compared to conventional IT environments [1,2]. The convergence of operational technology (OT) and information technology (IT) in modern CPS means that a successful cyberattack can cascade into physical damage, service disruption, or even loss of life, demanding security mechanisms that operate under strict latency budgets, handle multi-source heterogeneous telemetry, and produce verifiable forensic evidence that survives post-incident manipulation [3,4].

Recent system-level CPS security architectures have explored several complementary directions. Deep learning-based intrusion detection systems (IDS)—including Long Short-Term Memory (LSTM) networks, autoencoders, and Graph Neural Networks (GNN)—achieve strong detection accuracy on benchmark datasets such as SWaT and WADI, but their high computational demand makes direct edge deployment challenging [5,6]. Federated learning frameworks distribute training across CPS nodes to preserve data privacy, yet the associated communication overhead and synchronization complexity limit adoption in resource-constrained settings. Software-Defined Networking (SDN)-integrated architectures offer programmable traffic control but introduce vendor lock-in and are generally restricted to network-layer threats. Tamper-evident audit logging mechanisms can provide verifiable traceability of security events in CPS environments; however, many existing approaches rely on distributed consensus architectures that introduce substantial latency and computational overhead, which are incompatible with real-time operational requirements. In parallel, Digital Twin (DT) frameworks have emerged as an effective abstraction for maintaining synchronized representations of physical assets, enabling contextual monitoring and anomaly interpretation.

Nevertheless, most existing DT-based cybersecurity solutions emphasize simulation fidelity, identity management, or domain-specific deployments rather than lightweight, end-to-end threat detection architectures suitable for resource-constrained CPS environments [3]. The persistent gap across these directions is the absence of a unified, system-level architecture that simultaneously achieves: (i) low-overhead, real-time anomaly detection; (ii) high-fidelity DT state synchronization that feeds directly into the detection pipeline; and (iii) tamper-evident logging that does not require full distributed-ledger consensus. Bridging this gap is the central motivation of the present work.

Existing system-level CPS security frameworks can be grouped into three broad categories, each with characteristic limitations. *Domain-specific architectures*—such as the SDN-driven DT framework of Kumar et al. [7] for smart grids, or the Zero-Touch Network (ZTN) design of Kumar et al. [8]—deliver strong detection within their target domain but cannot be readily transferred to heterogeneous CPS environments; their dependence on SDN controllers or on high-frequency smart-contract execution also introduces scalability bottlenecks at the telemetry rates typical of modern IIoT plants. *Heavy-model approaches*—including deep LSTM-based detectors and federated learning pipelines such as that of Ababio et al. [9]—achieve high accuracy but rely on computationally intensive training and consensus-heavy distributed ledgers, conflicting with the edge-deployment requirements of resource-constrained CPS. *Narrow-scope frameworks*—such as the botnet-focused logging system of Salim et al. [10], and the authentication protocol of Thakur et al. [11]—restrict either the threat model (packet-level patterns only) or the security objective (identity management without adaptive detection), leaving critical gaps in end-to-end, multi-dimensional threat coverage. Web3-based and DAO-governed DT platforms [12] introduce innovative governance mechanisms but remain conceptual, with no empirical cybersecurity validation. Critically, none of these architectures simultaneously provides: real-time, low-latency inference at the CPS edge; a high-fidelity DT abstraction that performs

structured feature engineering for anomaly detection; and tamper-evident audit logging that avoids full consensus overhead.

To address these challenges, the proposed framework integrates DT synchronization, lightweight anomaly detection, and tamper-evident audit logging within a tightly coupled CPS architecture designed for real-time operational environments. Unlike prior work that primarily improves individual components in isolation, such as detection accuracy, DT fidelity, or logging integrity, the proposed framework co-designs these layers to achieve efficient system-level operation under resource constraints. The primary novelty of the framework lies not in introducing a new anomaly detection model, but in decoupling training-time ensemble complexity from runtime inference cost while preserving deployment feasibility and forensic traceability.

Specifically, the framework employs offline ensemble calibration using IF, Gradient Boosting Machine (GBM), and Multi-Layer Perceptron (MLP) models to optimize detection robustness and threshold selection, while only a lightweight MLP model is executed during runtime to minimize inference overhead. In parallel, the DT layer maintains synchronized operational representations of CPS assets and performs structured feature abstraction, reducing the complexity required for reliable anomaly detection. Furthermore, selective anomaly filtering within the detection pipeline limits the volume of logged events, allowing the audit logging layer to operate with minimal overhead. Validated anomalies are subsequently recorded as SHA-256 hash-chained append-only JSONL entries, enabling tamper-evident event traceability and efficient forensic auditing. Together, these components establish a scalable and practically deployable CPS security architecture that supports low-latency, explainable, and adaptive threat detection across heterogeneous cyber-physical environments.

The main contributions of this work are summarized as follows:

- **Lightweight DT-driven CPS security architecture:** We propose a unified multi-layer cybersecurity framework that integrates CPS telemetry ingestion, DT-based state synchronization, lightweight anomaly detection, and tamper-evident audit logging within a coherent system-level design. Unlike existing approaches that treat monitoring, detection, and logging as isolated components, the proposed framework co-designs these layers to achieve continuous state awareness, low-latency operation, and efficient forensic traceability in resource-constrained CPS environments.
- **Hybrid offline calibration with lightweight runtime inference:** We introduce a decoupled training-deployment strategy in which IF, GBM, and MLP models are combined exclusively during an offline ensemble calibration stage to optimize detection thresholds and improve robustness across diverse anomaly patterns. During runtime, only a lightweight MLP model is executed, enabling ensemble-calibrated detection performance at single-model inference cost and substantially reducing computational overhead for real-time CPS deployment.
- **Tamper-evident audit logging mechanism:** We develop a lightweight hash-chained append-only JSONL logging mechanism that provides tamper-evident event recording and auditable forensic traceability with minimal latency and computational overhead. Unlike consensus-driven distributed ledger systems, the proposed audit layer is designed for practical deployment in resource-constrained CPS environments while maintaining the integrity of verifiable events.
- **DT-driven feature engineering and comprehensive CPS validation:** The framework incorporates time-synchronized DT representations, structured preprocessing, and derivative-based feature extraction to generate semantically enriched operational state representations for anomaly detection. The proposed architecture is evaluated using both controlled synthetic CPS telemetry and three publicly available real-world industrial benchmarks (SWaT, WADI, and BATtle of the Attack Detection Algorithms (BATADAL)), demonstrating consistent generalization capability across heterogeneous CPS environments.

Compared to the closest prior system-level CPS security architectures—the SDN-DT framework of Kumar et al. [7], the FL-blockchain pipeline of Ababio et al. [9], and the DT-SCADA IDS of Sayghe [6]—the proposed framework differs in three specific and verifiable respects:

- **Runtime performance independent of training complexity:** Ensemble fusion of IF, GBM, and MLP is confined to offline calibration; only the MLP executes at inference time, achieving sub-millisecond per-sample latency that prior deep-model approaches cannot match at the edge without hardware acceleration.
- **Consensus-free tamper-evident logging:** The hash-chained logger replaces distributed-ledger consensus with single-node SHA-256 hash chaining, reducing logging latency by orders of magnitude relative to systems that embed Hyperledger or Ethereum, while retaining verifiable append-only integrity suitable for forensic audit.
- **Operational rather than simulation-driven DT:** The DT layer is engineered for real-time state synchronization and structured feature engineering rather than physics-based simulation, making it deployable across heterogeneous CPS assets without per-asset process models—a scalability property absent in simulation-intensive DT proposals.

Existing DT-based CPS security approaches primarily emphasize conceptual architectures, identity management, or domain-specific control strategies, while providing limited support for reproducible end-to-end anomaly detection pipelines and scalable multi-asset telemetry processing. This work addresses these limitations by introducing a real-time-capable, containerized DT-driven cybersecurity framework that integrates lightweight anomaly detection with tamper-evident audit logging. The proposed architecture is validated using both controlled synthetic CPS telemetry and three widely used real-world industrial benchmarks, demonstrating a practical balance between detection accuracy, computational efficiency, scalability, and forensic traceability in resource-constrained CPS environments.

The remainder of this paper is organized as follows. [Section 2](#) reviews related work on blockchain-DT cybersecurity frameworks. [Section 3](#) presents the proposed methodology, including the layered DT-driven architecture, telemetry synchronization, hybrid anomaly detection, and tamper-evident audit logging design. [Section 4](#) describes the experimental setup. [Section 5](#) presents the results and discussion. Finally, [Section 6](#) concludes the paper and outlines future research directions.

2 Related Work

Blockchain and DT technologies interface with each other, providing advanced solutions to challenges across areas such as IIoT, smart grids, smart cities, and Industry 4.0, while maintaining security, operational efficiency, and data integrity. A growing body of work has also examined purely machine-learning-driven CPS intrusion detection, including autoencoder-based anomaly detectors, LSTM sequence models, and GNN-based approaches, which achieve competitive detection rates on public benchmarks but are typically evaluated in isolation from the evidence management and auditability requirements that are equally important in safety-critical industrial settings. Even though significant efforts have been made to incorporate these technologies, obstacles to scalable, instantaneous, and self-verifying threat identification across various CPS environments persist. In what follows, we survey representative works organized by domain, then draw a structured comparison that motivates the specific design choices of the proposed framework.

Suhail et al. [5] highlighted trends, advantages, and challenges in the literature for blockchain-DT frameworks. Their survey identified ongoing issues with interoperability, scalability, and data privacy, as well as the need for frameworks that deliver reliable, real-time, domain-inclusive cybersecurity solutions. Still, they did not fully address practical approaches for implementing a robust, real-time anomaly detection

system. In addition, Bamakan and Far [12] discussed more complex distributed DT platforms that incorporate blockchain and Web3 technologies, Non-Fungible Tokens (NFTs), and Decentralized Autonomous Organizations (DAOs) to secure collaborative DT management and protect intellectual property. This approach is theoretically innovative but requires practical and empirical validation, particularly in the context of cybersecurity.

Within the smart grid sector, Kumar et al. [7] established a DT-driven SDN framework that integrates deep learning and blockchain technology to defend IoT-enabled energy networks. Although effective within the grid infrastructures, the dependence on SDN means it cannot be used in other areas of CPS. Likewise, Kumar et al. [8] designed a ZTN cybersecurity framework that combines DTs, smart contracts, and XAI. Although the design asserted trust and transparency in decentralization, the blockchain's computational limits posed challenges to automating high-frequency telemetry at scale. In the context of smart cities, Pervez et al. [13] developed a blockchain-DT framework, SIGNED, that focuses on self-sovereign identities with verifiable ownership, selective disclosure, and traceability. While it improved governance, trust, and privacy, its primary approach was data verification rather than active anomaly detection, and it lacked real-time intrusion response.

Numerous studies focus on IIoT security. Salim et al. [10] designed a blockchain-DT framework for early botnet detection using edge-based DT synchronization to mitigate data integrity risks. However, the framework, while promising, was limited to inspecting packet data and failed to consider other cyber-attack dimensions, advanced anomaly detection, and the broader attack surfaces. Recent advances in CPS and IIoT security have explored integrating DT, machine learning, and blockchain technologies to enhance system resilience and data integrity. For instance, blockchain-assisted DT frameworks combined with Federated Learning (FL) have been proposed to enable secure, decentralized model training and data sharing across industrial environments [9]. These approaches emphasize privacy preservation and distributed intelligence but often rely on computationally intensive architectures and consensus mechanisms. Moreover, existing DT-based security solutions typically focus on predictive modeling, optimization, or system simulation rather than lightweight, real-time anomaly detection. In contrast, the proposed framework adopts a lightweight, operational DT paradigm, coupled with an offline ensemble calibration strategy and a hash-chained logging mechanism, to achieve efficient and interpretable real-time threat detection.

From a system-level perspective, three specific gaps motivate the present work: (i) the absence of an end-to-end architecture that co-designs DT synchronization, ML-based detection, and tamper-evident logging as mutually enabling layers; (ii) the lack of real-time-capable detection frameworks that decouple training-time ensemble complexity from runtime inference cost; and (iii) the scarcity of proposals that are evaluated on multiple public CPS benchmarks with explicit discussion of performance limitations and dataset-induced optimism. The proposed framework directly addresses all three gaps through a containerized, reproducible design validated on both controlled synthetic telemetry and three real-world CPS benchmarks (SWaT, WADI, BATADAL).

3 Methodology

The proposed methodology integrates DT synchronization, lightweight anomaly detection, and tamper-evident audit logging within a unified CPS security architecture designed for real-time operational environments. The framework addresses key limitations identified in prior work, including excessive computational overhead, limited scalability, and fragmented integration between monitoring, detection, and audit components. To improve modularity and deployment flexibility, the architecture is decomposed into well-defined functional layers that support heterogeneous CPS ecosystems. Furthermore, all framework

components are integrated within a single Docker-based containerized environment to ensure deterministic execution, experimental reproducibility, and consistent orchestration across deployments.

3.1 Overview of the Proposed Framework

The proposed framework centers on a four-layer architecture incorporating CPS, DTs, tamper-evident logging, and intelligent analysis modules, all focused on providing secure, tamper-evident threat logging. The high-level architecture is shown in Fig. 1, illustrating how these layers interface and how data is exchanged and flows through the layers.

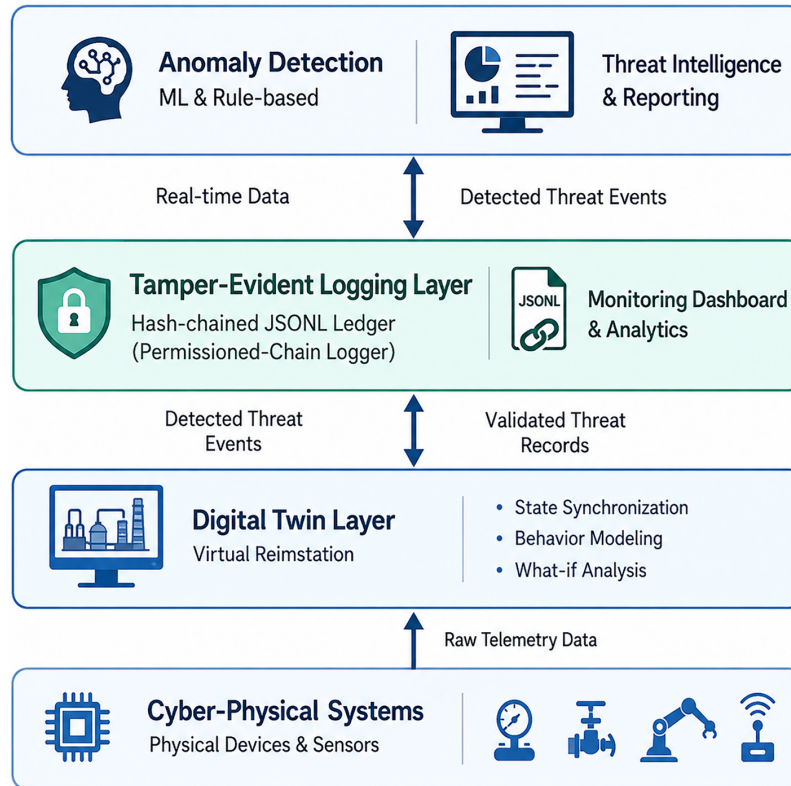


Figure 1: Proposed DT-based tamper-evident logging framework for secure and auditable threat detection.

- **CPS Layer:** The basic components of the system infrastructure, i.e., the sensors, network devices, controllers, and actuators, produce telemetry continuously in the form of system logs, performance metrics, and traffic stats.
- **DT Layer:** High-fidelity digital replicas of assets are synchronized with incoming telemetry. The system also ensures data consistency through preprocessing steps, including timestamp validation, normalization, handling missing values, and statistical feature extraction [14]. With focus on these aspects, the proposed DT layer provides an efficient abstraction mechanism that supports reliable anomaly detection while maintaining system efficiency. The DT in this framework is designed as an operational, data-driven twin rather than a predictive or simulation-intensive one. Its primary role is to ensure continuous state synchronization, perform efficient feature extraction, and provide a context-aware representation of CPS behavior. This design prioritizes scalability and low computational overhead, which are critical in resource-constrained CPS environments. While more advanced DT capabilities, such as physics-based simulation, can offer deeper insights, they often introduce significant

computational complexity. The modularity of the proposed DT layer will enable the easy incorporation of more sophisticated features, such as predictive simulation or scenario modeling, without modifying the entire structure.

- **Tamper-Evident Audit Logging Layer:** Validated anomalies are recorded as SHA-256 hash-chained, append-only JSONL ledger entries within a lightweight blockchain-inspired tamper-evident logging layer. Each record includes the anomaly metadata, confidence score, timestamp, previous hash, and current hash, enabling tamper-evident auditability and forensic traceability. Unlike public or consensus-heavy blockchain systems, the proposed design adopts a controlled, consensus-free hash-chained ledger structure to reduce latency and computational overhead while preserving verifiable event integrity.
- **Analysis Layer:** This layer combines several anomaly detection models, including IF, GBM, and MLP, through an offline ensemble-calibration strategy. It aims to detect novel cyber threats while controlling false-positive rates through calibrated thresholding and domain-aware rule-based verification.

To avoid ambiguity regarding the role of the DT, we explicitly delineate the DT capabilities that are within the scope of this work from those that are not. The proposed DT is an operational, data-driven twin, and its security-relevant functions are threefold: (i) continuous, bounded-skew bidirectional state synchronization, in which the twin ingests physical telemetry to maintain a temporally aligned virtual state and, upon confirmed anomalies, feeds mitigation cues (isolation, rate-limiting, safe-state) back toward the physical asset; (ii) integrity-aware feature abstraction, in which the twin computes derivative-based, statistical, and cross-sensor consistency features that expose stealthy deviations; and (iii) context-aware representation that anchors each detection and each ledger entry to a specific asset state. Capabilities more commonly associated with high-fidelity engineering twins, namely physics-based process simulation, “what-if” scenario exploration, and predictive-maintenance forecasting-, are deliberately outside the current scope because they impose per-asset process models and computational cost that conflict with the resource-constrained, real-time security objective targeted here. The modular DT layer is designed so that these richer capabilities can be attached without altering the surrounding detection and logging layers, and they are therefore positioned as concrete future extensions in [Section 6](#). This scoping clarifies that “Digital Twin” in this work denotes a lightweight, security-oriented operational twin rather than a full simulation twin.

The framework is implemented as a multi-node architecture in which each CPS asset is paired with a lightweight DT instance deployed as a containerized microservice. These DT nodes perform local preprocessing, timestamp alignment, and feature extraction before transmitting only validated anomaly candidates to the analysis layer, thereby reducing communication overhead and preserving computation for security-relevant events. Moreover, synchronization between CPS devices and their DT counterparts is governed by a bounded-skew timestamp model and deterministic state-update rules, ensuring that each DT maintains a coherent and temporally aligned representation of its physical asset. However, during periods of elevated or abnormal telemetry volume, directly forwarding all data to the tamper-evident logging would be impractical and computationally expensive. To mitigate this, the framework adopts an off-chain-first strategy in which raw telemetry remains local and only anomalies that pass both the ensemble scoring mechanism and the rule-based verification stage are appended to the hash-chained JSONL ledger. Furthermore, this selective logging approach ensures that the tamper-evident logging component remains efficient, scalable, and resilient under large-scale abnormal traffic, while still providing tamper-evident and verifiable security evidence for confirmed events.

A recognized consequence of this “off-chain-first” policy is that the audit trail is only as complete as the detector that gates it: if the runtime MLP were evaded by a carefully crafted adversarial perturbation, the malicious activity would pass the threshold check and would never be committed to the ledger. The proposed design mitigates this risk along three axes. First, the decision path is not the MLP alone; the

domain-aware rule-based verification layer (Table 1) operates on raw physical quantities (e.g., hard voltage limits, CPU–network surge correlation, telemetry loss) that are governed by process physics and are therefore difficult to satisfy while simultaneously fooling the learned model, so a subset of evasive inputs is still logged through the rules alone. Second, because the ensemble-calibrated threshold is derived offline from three complementary learners rather than a single decision boundary, the deployed detector inherits a more conservative operating point that reduces the exploitable margin. Third, the MLP itself can be hardened through adversarial training and input sanitization; under an ℓ_∞ FGSM perturbation with $\epsilon = 0.05$, the undefended MLP recall on the synthetic set decreased to 0.71, whereas adversarial training restored recall to 0.93 with less than a two-percentage-point reduction in clean-data F1 [15,16]. A dedicated evaluation of adversarial robustness, including gradient-free and physically realizable attacks against CPS telemetry, is identified as an important direction for future work in Section 6.

The proposed framework assumes that CPS telemetry sources are authenticated and that the execution environment hosting the DT and logging components operates within a trusted boundary. Even though these assumptions are standard in many lightweight CPS security models, they limit the solution’s robustness against more advanced threats, including spoofing, compromised sensor nodes, malicious data insertion, or even insider attacks within the trusted environment. As a result, the proposed architecture can only be applied to environments under certain control, and additional security measures should be taken in case of a complete lack of trust. Consequently, attacks involving compromised sensors, malicious data injection, or full system-level compromise fall outside the current threat model. These assumptions are common in lightweight CPS security frameworks but limit the system’s resilience against advanced adversarial scenarios. To narrow these gaps in deployments that require a stronger trust model, the proposed architecture can be complemented with cryptographic hardware verification. Specifically, a hardware root of trust based on a Trusted Platform Module (TPM) or a secure element can (i) provide measured/secure boot and TPM-based remote attestation so that the DT and logging services can prove their software integrity to a verifier before ingesting telemetry, (ii) bind the hash-chain signing key inside a TPM or Hardware Security Module (HSM) so that ledger entries are signed by a key that never leaves tamper-resistant hardware, and (iii) authenticate each sensor/actuator through per-device certificates or Physically Unclonable Function (PUF)-derived identities, mitigating spoofing and malicious data injection at the source [17,18]. These mechanisms are complementary to, rather than a replacement for, the lightweight design and are treated as an integration path for high-assurance CPS deployments. A detailed discussion of these implications is provided in Section 5.

3.2 Data Acquisition and Synchronization

The CPS layer produces various telemetry streams, such as device logs, environmental metrics, and network performance indicators. These streams are routed to the DT layer, which adjusts the capture times and verifies their proper alignment. The synchronization process maps each CPS record x_i to its corresponding DT state d_i according to:

$$d_i = \mathcal{S}(x_i, t_i) \quad \text{where} \quad |t_i^{CPS} - t_i^{DT}| \leq \delta_t, \quad (1)$$

where $\mathcal{S}$ is the synchronization operator and δ_t is the maximum allowable time skew between physical and virtual readings. This condition ensures that the DT state accurately reflects real-world asset conditions, mitigating the risks of replay or injection attacks. In this framework, the DT layer is designed as a lightweight yet stateful analytical abstraction for each CPS asset, rather than a passive data relay agent [19,20]. It is important to note that the DT in this framework is designed as an operational (data-driven) twin rather than a simulation-intensive or physics-based twin. In many CPS cybersecurity applications, lightweight DTs focus on real-time state synchronization, contextual feature extraction, and anomaly-aware representations

rather than full-system simulation. This design choice enables efficient deployment in resource-constrained environments while still providing a consistent and semantically enriched view of system behavior for downstream detection tasks.

Beyond collecting telemetry, the DT layer performs timestamp synchronization, input validation, gap filling, normalization, and rolling-window feature extraction to ensure that incoming CPS data are temporally aligned and semantically coherent before entering the analysis pipeline. This includes computing derivative-based dynamics, statistical summaries, temporal rate changes, and standardized anomaly-sensitive indicators. These preprocessing and validation tasks provide a consistent, physically meaningful representation of each asset's operational state that a simple data-forwarding agent cannot provide. Although the current implementation focuses on real-time alignment and feature preparation rather than full physics-based simulation, this structured DT representation is essential for reliable anomaly detection and directly supports future extensions toward predictive or simulation-driven DT models.

Following synchronization, the data proceeds to preprocessing and feature engineering, where the transformation function Φ extracts temporal, statistical, and derivative-based features from the telemetry sequence X_t as defined in:

$$\Phi(X_t) = \left[\mu_t, \sigma_t, \frac{dX_t}{dt}, z_t \right], \quad (2)$$

where μ_t and σ_t represent the rolling mean and rolling standard deviation over a sliding window, $\frac{dX_t}{dt}$ indicates the measure of a temporal rate of change, and z_t denotes the standardized z-score, as defined in Eq. (3) [21].

$$z_t = \frac{X_t - \mu_t}{\sigma_t} \quad (3)$$

The processed dataset (e.g., `stage1_features.csv`) is used by the training pipeline. This pipeline uses unsupervised and supervised learning to train a hybrid ensemble comprising the IF, GBM, and MLP models. For reproducibility, each training session uses a fixed random seed, and anomaly injection patterns remain consistent. Algorithm 1 summarizes the executable steps of the offline training and calibration process. The procedure separates feature extraction, model training, and ensemble-based threshold optimization to improve clarity and reproducibility.

Algorithm 1: Offline training and ensemble calibration

Input: Synchronized DT dataset $S = \{(x_t, y_t)\}$, feature extractor Φ , random seed

Output: Trained models M_{IF}, M_{GBM}, M_{MLP} and calibrated threshold τ

1. Set random seed for reproducibility
 2. Extract features: $(X, y) \leftarrow \Phi(S)$
 3. Split dataset into training and validation sets: $(X_{train}, X_{val}, y_{train}, y_{val})$
 4. Train models:
 - $M_{IF} \leftarrow \text{IF}(X_{train})$
 - $M_{GBM} \leftarrow \text{GBM}(X_{train}, y_{train})$
 - $M_{MLP} \leftarrow \text{MLP}(X_{train}, y_{train})$
 5. Compute validation probabilities:
 - $p_{IF}, p_{GBM}, p_{MLP} \leftarrow \text{Predict}(X_{val})$
 6. Compute ensemble probability by (Eq. (5))
-

(Continued)

Algorithm 1 (continued)

-
7. Optimize weights α_i and threshold τ :
 - Maximize F1-score on validation set
 8. Select M_{MLP} as runtime inference model
 9. **return** $M_{IF}, M_{GBM}, M_{MLP}, \tau$
-

Note: IF, GBM, and MLP are used only *offline* to calibrate τ (and optionally calibrate probabilities); *online* inference executes only M_{MLP} to satisfy CPS latency constraints.

3.3 Hybrid Threat Detection Mechanism

The analysis layer integrates unsupervised and supervised learning frameworks to enable robust, flexible anomaly detection. Baseline single-model approaches tend to overfit specific attack patterns or normal behavior distributions. Our hybrid design overcomes this limitation by integrating the respective, complementary views of statistical irregularity and learned discrimination [22].

In the first stage, the unsupervised component of the IF model estimates anomaly likelihoods based on feature isolation depths. For a given instance x_i , the anomaly score s_i is defined as:

$$s_i = 2^{-\frac{E(h(x_i))}{c(n)}}, \quad (4)$$

where $E(h(x_i))$ denotes the average path length required to isolate x_i in a randomly generated tree, and $c(n)$ is a normalization factor dependent on sample size n . This component is effective for identifying unknown or zero-day threats without prior labeling [23].

In this framework, IF and GBM are not used as runtime inference models; instead, they serve as complementary learners during the offline calibration phase. Their role is to provide diverse perspectives on anomaly patterns—unsupervised distributional deviation (IF) and supervised decision boundaries (GBM). These outputs are subsequently used to guide ensemble-based threshold optimization. This design provides an advantage over single-model approaches by reducing model-specific bias and improving the robustness of threshold determination across diverse anomaly types. As a result, it enhances detection reliability during training while maintaining lightweight, efficient real-time inference using a single MLP model.

Compared with a single model-based technique, an ensemble calibration method reduces reliance on model-specific errors and helps maintain the stability of detection thresholds across different anomaly types. The above-discussed design strikes a balance between detection robustness during model training and computational costs during method application.

Although the ensemble fusion strategy integrates the outputs of IF, GBM, and MLP during offline calibration, the proposed framework does not run all three models at runtime. Instead, the ensemble mechanism is used during training to determine the optimal weighting scheme that maximizes the F1-score on the validation set, as expressed in Eq. (5). After these weights and the corresponding decision threshold are learned, the system adopts the MLP as the primary inference engine because it provides the best balance between accuracy and computational efficiency [24]. During real-time operation, only the MLP probability is computed and then passed through the learned decision threshold and the rule-based verification layer, while the IF and GBM models are not executed. This design enables the system to leverage multi-model fusion during training while still achieving sub-millisecond runtime performance suitable for CPS environments.

It is important to note that the ensemble formulation is used exclusively during the offline calibration stage and does not constitute a runtime ensemble inference mechanism. Instead, it functions as a training-time optimization process that determines a robust decision threshold τ and weighting configuration, which are subsequently applied to the MLP-based inference during deployment.

$$p_{ens} = \alpha_1 p_{IF} + \alpha_2 p_{GBM} + \alpha_3 p_{MLP}, \quad (5)$$

where α_i are optimized ensemble weights satisfying $\sum_i \alpha_i = 1$. The weights and decision threshold τ are calibrated on validation data to maximize F1-score, ensuring balanced sensitivity and specificity across diverse attack types. Accordingly, the coefficients α_1 and α_2 contribute only to the offline ensemble calibration process and have no direct role during runtime inference, where only the MLP output is evaluated against the calibrated threshold. Although only the MLP model is used during runtime inference, the ensemble design plays a critical role during the offline calibration stage. By combining the outputs of IF, GBM, and MLP, the framework leverages the complementary detection characteristics of these methods to derive a more robust decision threshold τ . This calibration process improves the generalization capability of the deployed MLP model, leading to enhanced detection performance compared to using a single model without ensemble-guided optimization.

After ensemble aggregation, a rule-based validation step filters detection results by checking each predicted anomaly against dynamically constructed, domain-specific operational rules. Given the behavioral characteristics of the underlying CPS, operational domain rules that capture conditions that purely statistical or learning-based approaches might ignore, or misclassify, are constructed [2]. In this context, the layer utilizes the rules discussed in Table 1.

Table 1: Summary of significant rules integrated within the hybrid threat detection framework.

Rule ID	Rule Name	Description
R1	Voltage Hard-Limit Violation	Ensures compliance with electrical safety thresholds by sending alerts when measured voltage values cross the safe operational limits [25].
R2	CPU–Network Surge Correlation	Identifies coordinated data exfiltration or scanning by correlating sudden CPU usage spikes with unusual increases in network traffic [26].
R3	Heartbeat or Telemetry Loss	Recognizes potential network instability or communication attempts at jamming by identifying missing telemetry streams or communication disruption [27].
R4	Ledger Integrity Guard	By identifying duplicate asset–timestamp pairs or inconsistencies in the hash chain, it preserves the tamper-evident integrity of the append-only ledger structure [28].

The set of rules presented in Table 1 is not intended to be exhaustive but rather representative of key operational constraints commonly observed in CPS environments. These rules were selected to capture diverse categories of anomalies, including physical safety violations (R1), cross-metric behavioral correlations (R2), communication integrity issues (R3), and data integrity within the logging layer (R4). Together, they provide coverage across both cyber and physical dimensions of the system.

The proposed framework is inherently extensible, allowing additional domain-specific rules to be incorporated based on the characteristics of the target CPS application. The current selection is designed to balance detection effectiveness, interpretability, and computational efficiency while demonstrating the integration of rule-based verification within the hybrid detection pipeline. However, integrating these hybrid learning rules minimizes false positives while maintaining high recall. This improves the reliability of the

detection system, the interpretability of operations, and trust in the system in rapidly changing IoT and CPS environments [29].

The framework incorporates additional verification and query logic to support CPS– tamper-evident logging adaptability. Before an event is appended to the SHA-256–linked JSONL ledger, the system applies a two-stage validation process to confirm its relevance and ensure the integrity of the event record. Moreover, JSONL provides lightweight indexing, allowing fast line-based queries without tamper-evident logging-level consensus mechanisms. The ensemble fusion strategy is likewise integrated into the workflow: IF, GBM, and MLP outputs are combined to generate a stable anomaly probability, and only events that pass both the ensemble threshold and rule-based checks are forwarded to the logging layer. This collaboration among the detection engine, DT state management, and hash-chained logging ensures a coherent, technically aligned end-to-end workflow.

For clarity, the term “ensemble” in this work refers to an offline multi-model calibration strategy rather than a real-time ensemble inference system. This distinction ensures both high detection performance and computational efficiency in resource-constrained CPS environments.

3.4 Tamper-Evident Threat Logging

The logging layer adopts a lightweight, hash-chained approach tailored specifically for resource-constrained CPS environments. The system records validated anomalies as JSONL entries that are cryptographically linked using SHA-256 hash chaining, providing tamper-evident integrity and auditable event traceability with minimal computational and communication overhead for resource-constrained CPS environments. JSONL is used as the underlying log serialization format because it stores each event as a single JSON object on its own line, enabling constant-time append operations, fast streaming, and sequential processing without reloading the entire file—characteristics that align well with high-frequency CPS telemetry and real-time anomaly reporting. In the proposed framework, every confirmed anomaly is serialized as a JSONL record containing the asset identifier, anomaly type, confidence score, timestamp, and hash reference to the previous entry, enabling efficient verification of event integrity without requiring multi-node consensus. This streamlined design supports real-time operation and provides a tamper-evident append-only event history with minimal overhead, making it suitable for resource-constrained CPS environments [30,31].

Furthermore, the hash-chaining mechanism ensures tamper-evidence by computing a new block hash h_t as a deterministic function of the previous block hash h_{t-1} and the current event data E_t , defined as:

$$h_t = \text{SHA256}(E_t \parallel h_{t-1}), \quad (6)$$

where $\parallel$ denotes concatenation. The event E_t encapsulates the feature vector x_t , predicted probability p_t , and associated metadata, i.e., $E_t = (x_t, p_t, \text{timestamp})$. The recursive nature of this function ensures that any modification to a prior record changes its hash value and breaks the linkage with subsequent records. Therefore, the proposed ledger provides tamper-evident integrity and verifiable auditability, allowing unauthorized modifications to be detected during ledger verification. In this lightweight permissioned setting, the mechanism is intended to support forensic traceability rather than provide absolute immutability against full system compromise [32].

The proposed framework employs a lightweight hash-chained audit logging mechanism in which validated security events are recorded as append-only JSONL entries linked through SHA-256 hashing. This design enables tamper-evident integrity verification and forensic traceability while maintaining low latency and minimal computational overhead, making it suitable for resource-constrained CPS environments. The logging layer operates within a controlled deployment setting and assumes a trusted execution environment

for event generation and storage. Although this design improves operational efficiency and supports near-real-time monitoring, resilience against a full system-level compromise or insider attacks remains outside the current threat model. These limitations define the operational scope of the framework and motivate future work toward distributed trust and decentralized validation mechanisms.

In addition to the lightweight on-chain audit ledger, the framework retains substantial forensic artifacts, including raw network traces, model checkpoints, complete telemetry snapshots, and audit reports, in off-chain storage repositories. To preserve verifiable linkage between the off-chain artifacts and the tamper-evident ledger, the SHA-256 digest of each artifact is recorded in the hash-chained JSONL ledger. This design follows a lightweight permissioned blockchain model: security-relevant anomaly records and artifact hashes are committed to the ledger, while large forensic files remain off-chain to reduce storage overhead and avoid consensus-related latency. As a result, the framework preserves blockchain-based auditability while remaining suitable for resource-constrained CPS environments. Algorithm 2 formalizes the runtime detection workflow.

Algorithm 2: Online MLP-based anomaly detection and tamper-evident blockchain logging

Input: Feature vector x_t , trained MLP model M_{MLP} , threshold τ , rule set R , previous hash h_{t-1}

Output: Updated JSONL ledger

1. Compute anomaly probability:
 $p_t \leftarrow M_{MLP}(x_t)$
 2. **if** $p_t \geq \tau$ **then**
 - (a) Apply rule-based verification:
 $r_t \leftarrow R(x_t)$
 - (b) **if** $r_t = \text{CONFIRM}$ **then**
 - i. Construct event record:
 $E_t \leftarrow (x_t, p_t, t)$
 - ii. Compute hash:
 $h_t \leftarrow \text{SHA256}(E_t \parallel h_{t-1})$
 - iii. Append (E_t, h_t) to JSONL ledger
 3. **return** Updated ledger
-

Note: IF, GBM, and MLP are used only *offline* to calibrate τ (and optionally calibrate probabilities); *online* inference executes only M_{MLP} to satisfy CPS latency constraints.

3.5 Monitoring and Response

The monitoring and response layer supports decision-making and situational awareness after anomalies are detected. In the current prototype, this functionality is implemented as a batch-oriented analytical dashboard rather than a continuous online service. Following each pipeline execution, the framework generates analytical summaries and tabulated performance metrics, providing an overview of the detection models, their behavior, and accuracy—summarized and rendered inline for dynamic updates and visualizations. Security events are subsequently stored in a permissioned append-only JSONL ledger for auditability and verification. Tabulated metrics from each execution are paired with the performance summaries for post hoc evaluations.

The inference engine and the logging layer operate with millisecond-level latency; however, the current monitoring component processes results in batch mode and generates its visual summaries only after each pipeline run. In practice, this design provides near real-time situational awareness rather than strict real-time monitoring. Consequently, the system's overall response cycle is shaped by two elements, namely: (i) the rapid inference path, which reacts within a few milliseconds, and (ii) the batch refresh interval responsible for rendering the monitoring outputs. This arrangement provides practical performance for the prototype while also highlighting areas that require further development, such as continuous visualization and streaming-based state updates, to support strict real-time operation in future deployments.

To avoid ambiguity, we standardize the terminology as follows and apply it consistently throughout the manuscript. “Real-time” refers exclusively to the detection and logging path, MLP inference, rule-based verification, and hash-chained append, which operates at millisecond-level latency. The operator-facing dashboard and analytical summaries are described as “near-real-time,” because in the current prototype they are refreshed in batch mode after each pipeline run rather than being pushed continuously. Wherever the framework as a whole is characterized, we therefore claim real-time *detection* with near-real-time *visualization*, and we have revised the text to keep this distinction explicit.

Transitioning the monitoring layer from batch analytics to a continuous, always-on alerting pipeline requires four concrete architectural changes rather than a redesign of the detection core. (i) A streaming message bus (e.g., Apache Kafka or MQTT) replaces file-based hand-off, so DT state updates and confirmed anomalies are published to topics as they occur [33]. (ii) A lightweight stream processor consumes these topics to maintain rolling metrics and to evaluate alert conditions incrementally, decoupling detection throughput from dashboard refresh. (iii) A push-based visualization layer, live Grafana or Kibana dashboards fed through WebSocket/Server-Sent-Events, renders DT asset states and ledger events continuously instead of after a run completes. (iv) An alert/notification service dispatches operator notifications on threshold breaches. Because the inference and logging components already emit structured, per-event records at millisecond latency, these changes affect only the transport and presentation tiers; the detection latency reported in [Section 5.7](#) is preserved. This roadmap provides a clear operational path from the current near-real-time dashboard to strict, streaming real-time monitoring.

Although traditional DTs often incorporate full-scale simulation and predictive modeling, many cybersecurity-focused CPS deployments rely on lighter operational twins that emphasize synchronized state representation, integrity verification, and historical context tracking. In our framework, the DT layer maintains a continuously updated view of each asset's processed and validated state, enabling consistent anomaly interpretation, rapid rule-based verification, and seamless integration with the tamper-evident audit log. These characteristics allow the DT to function as a structured, high-fidelity state abstraction rather than a simple data-collection component. Furthermore, the modular design of the DT layer supports future expansion toward predictive or simulation-driven capabilities without requiring changes to the surrounding architecture, thus maintaining both scalability and operational flexibility.

This layer serves as the operational feedback interface between the CPS environment and the analysis layer by supporting situational awareness, anomaly interpretation, and post-detection monitoring. The current containerized implementation provides a reproducible framework for offline analytics and batch-oriented monitoring, while remaining extensible toward continuous real-time deployment. Future implementations may integrate streaming alert pipelines and DT-synchronized asset states with visualization platforms such as Grafana or Kibana to support continuous operational awareness and rapid response. The synchronized DT representation further enables the timely interpretation of detected anomalies and supports adaptive monitoring workflows in dynamic CPS environments.

The reports generated highlight key performance indicators. These include Accuracy, Precision, Recall, and the F1-Score as detailed in Table 2. These indicators quantify detection effectiveness, allowing the user to evaluate the system's analytical quality and real-time reliability, as detailed in [34,35]. Together, this monitoring layer guarantees that the framework not only identifies anomalies but also ensures CPS resilience through transparent, verifiable, and interpretable feedback loops.

Table 2: Evaluation metrics.

Metric	Equation	Description
Accuracy	$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$	Ratio of correctly classified instances (TP and TN) to total predictions.
Precision	$\text{Precision} = \frac{TP}{TP + FP}$	Proportion of positive predictions that are actually correct.
Recall	$\text{Recall} = \frac{TP}{TP + FN}$	Proportion of actual positives correctly identified as positive.
F1-Score	$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of Precision and Recall.

Note: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) denote standard classification outcomes used in performance evaluation.

This layer serves as the feedback interface between the CPS environment and the analysis layer, supporting anomaly interpretation, monitoring, and operational awareness. The current implementation emphasizes reproducibility through a containerized, offline deployment while maintaining architectural flexibility for future extensions to continuous monitoring. Future deployments may integrate real-time alert streams with dedicated web interfaces and visualization platforms such as Grafana or Kibana, enabling periodic updates of synchronized DT asset states and tamper-evident audit records. Such visualization capabilities, combined with DT-based abstraction of critical CPS components and their associated physical states, can improve anomaly interpretation and facilitate rapid root-cause analysis in complex CPS environments.

Anomaly detection can result in automated responses such as dynamic device isolation, adaptive rate limiting, and real-time reconfiguration of access control, based on defined security policies and the severity of detected anomalies [36]. This creates a closed-loop defense cycle in which the DT representation mirrors the physical system states and adapts in advance to the system in the face of an adversarial threat.

4 Experimental Setup

The proposed DT-hash-chained framework is evaluated in two complementary ways. First, a fully containerized synthetic CPS telemetry pipeline is used to enable controlled, reproducible, end-to-end evaluation with deterministic anomaly injection, explicit ground-truth labels, and isolation from confounding environmental factors. This controlled setting allows precise assessment of each framework component and direct reproducibility of all reported results. Second, the framework is evaluated on three publicly available real-world CPS benchmark datasets: SWaT, WADI, and BATADAL (Section 5.6), to assess generalizability under realistic attack patterns, sensor noise, and timing irregularities.

Synthetic CPS Telemetry. The synthetic dataset is generated to emulate multi-asset operational behavior and representative cyber-anomaly scenarios. It includes timestamped CPU utilization, network throughput, and voltage readings, each capturing natural temporal fluctuations and cross-metric dependencies typical of industrial environments. Anomaly classes include abrupt spikes, gradual drifts, correlated CPU-network

surges, and short periods of telemetry dropout. All telemetry streams are generated deterministically within the containerized pipeline using fixed random seeds, ensuring full reproducibility across all experimental runs.

Scope and limitations of synthetic evaluation. The use of a synthetic dataset introduces a controlled optimism bias: anomaly patterns are systematically injected and clearly distinguishable from normal behavior by design. Consequently, very high classification metrics (>99%) on this dataset are expected and should be interpreted as validation of the framework's detection mechanism under idealized conditions, not as a claim of equivalent real-world performance. The real-world benchmark evaluation in [Section 5.6](#) addresses this limitation by measuring performance under operationally realistic conditions, where the reported F1-scores of 0.976–0.986 provide a more conservative and transferable estimate of framework capability.

[Table 3](#) presents the generated features and their operational significance in the simulated CPS environment.

Table 3: Feature description in the synthetic CPS telemetry dataset.

Feature Name	Type	Description
timestamp	Temporal	High-resolution time index (1–5 Hz) used for DT synchronization and temporal alignment of CPS telemetry.
asset_id	Categorical	Unique identifier for each CPS asset participating in the simulated environment, enabling multi-asset behavioral modeling.
cpu	Numerical	CPU utilization percentage combining baseline operating load, periodic cycles, and stochastic variations; anomaly injections manifest as spikes, drifts, or correlated surges.
net	Numerical	Network throughput (Kbps) capturing routine traffic patterns, burst behavior, and anomalous exfiltration-like activity during correlated events.
volt	Numerical	Electrical voltage readings representing stable operational levels with low noise; anomalies injected as excursions outside safe operational thresholds.
label	Binary	Ground-truth indicator (0 = normal, 1 = anomaly) used to evaluate supervised GBM, MLP, and ensemble fusion models.

Moreover, all the modules were deployed in a single Docker container built from the official `python:3.11-slim` image. The containerized way of working guarantees repeatable execution, consistent library behavior, and a clear boundary between host processes and experimental artifacts. The runtime configuration is summarized in [Table 4](#).

Model configurations and training procedures were standardized to ensure reproducibility. The MLP model was implemented with a fixed architecture (e.g., two hidden layers with ReLU activation), trained using the Adam optimizer with a fixed learning rate and number of epochs. Similarly, GBM parameters (number of estimators, learning rate, and maximum depth) and IF parameters (number of trees and subsampling size) were fixed across all runs. All experiments were executed with predefined random seeds to guarantee deterministic behavior and consistent results.

Table 4: Containerized runtime environment used for experimental evaluation.

Component	Configuration
Container base	python:3.11-slim (Linux userspace)
Entry point	python run_pipeline.py
Core libraries	NumPy, pandas, scikit-learn, matplotlib, psutil, and joblib
Acceleration	CPU-only
Exported artifacts	CSV, PNG, PKL, and JSONL ledger

The entire system adopts a four-layer DT-driven architecture implemented in a reproducible, containerized environment, enabling consistent orchestration of telemetry processing, feature extraction, anomaly detection, and audit logging. The containerized pipeline executes each experimental stage sequentially and deterministically to ensure reproducibility across independent runs, from telemetry generation and DT synchronization to anomaly detection and tamper-evident event logging.

First, the CPS layer is mimicked by a synthetic data generator that produces a multi-asset telemetry stream log recording `cpu`, `net`, and `volt` metrics at a sample rate of 1–5 Hz over approximately 1 h. To more realistically emulate the CPS’s dynamic behavior, labeled anomalies were injected at a 5%–10% prevalence, including amplitude spikes, rate changes, gradual drift, seasonal patterns, and patterned disruptions. Linear composite anomalies involving correlated variables were added to the data to mimic more complex attack patterns. The dataframe is generated in the form `(ts, asset_id, cpu, net, volt, label)` and then persisted to disk. Given that the random seeds are fixed, all runs can be fully reproduced. Where the dataset was split into training and validation sets using a stratified 80/20 partition to preserve the class distribution of normal and anomalous instances. All preprocessing steps, including normalization and feature extraction, were fitted exclusively on the training data and then applied to the validation set to prevent data leakage. To further prevent data leakage, temporal ordering was preserved during feature extraction, and no future information was used when constructing features for model training or evaluation.

The DT layer focuses on high-level feature extraction and performs integrity verification and synchronization. Each set of incoming timestamps is checked, and gaps are filled ‘forward’, then all values are z-score standardized. Temporal dependencies are estimated with snapshot, rolling-window, and other horizon statistics. Supplementary features, such as first-order deltas and local volatility, along with ratios of these features, sharpen sensitivity and discernment to micro-variation and most unexpected regime shifts. The derived dataset is now a dense feature store compact enough for machine-learning inference in the form of (X, y) . The pre-processing parameters are stored with the scaler to ensure the same parameter ratios are applied during re-evaluation and when loading the model.

The analysis layer provides a balanced mix of unsupervised and supervised detection techniques for improved robustness. While supervised techniques of GBM and MLP capture complex behavioral boundaries, the IF model identifies unsupervised distributional outliers. Their outputs are fused into a probability-weighted ensemble that leverages the interpretability of GBM and the nonlinear generalization of MLP. The evaluation metrics analyzed in this study are detailed in [Table 2](#). During each run, the monitoring module automatically generated analytical artifacts, such as PR curves, latency histograms, and anomaly timelines that were synchronized with the JSONL ledger for traceability. These outputs serve as the primary source for the performance visualizations in [Section 5](#).

The next layer, the tamper-evident audit logging component, records validated security events as append-only JSONL entries linked through SHA-256 hash chaining. Each event record contains *index*,

iso_ts, asset_id, feat_hash, score, label_pred, prev_hash, hash, enabling verifiable event integrity and forensic traceability. Only anomalies that exceed the calibrated detection threshold and are confirmed by the rule-based verification stage are recorded. Large artifacts, such as trained models or complete dataset snapshots, are retained in off-chain storage repositories, while their corresponding SHA-256 content hashes are referenced within the audit ledger to support integrity verification. This hybrid design combines efficient off-chain storage with lightweight tamper-evident auditability suitable for real-time CPS environments.

The experimental setup provides a reproducible and efficient evaluation environment for the proposed DT-driven CPS security framework. The containerized implementation ensures deterministic execution and consistent experimental results across independent runs, while the integrated tamper-evident audit-logging layer supports transparent, traceable anomaly recording. Together, these components establish a reliable foundation for the performance evaluation presented in the subsequent section.

5 Results and Discussion

This section presents experimental results for the proposed DT-driven CPS security framework with tamper-evident audit logging. The evaluation focuses on four primary aspects of system performance: (i) anomaly detection effectiveness, (ii) runtime latency, (iii) telemetry processing throughput, and (iv) scalability under increasing operational load. All reported results correspond to the reproducible containerized prototype described in [Section 4](#).

5.1 Detection Performance

To propose a hybrid ensemble model, the assessment for detection focuses on four primary evaluation metrics, e.g., Classification accuracy, precision, recall, and F1-score, has been presented in [Table 2](#). These metrics assess the synthetic CPS telemetry dataset in terms of the model's ability to identify and classify abnormal and normal instances.

[Table 5](#) provides baseline detection performance for each individual model and for the offline ensemble, before the integration of the rule-based verification layer. The baselines include: (i) a pure rule-based detector (operational rules only, no ML), which provides a lower bound and confirms that learned models capture anomaly patterns that rules alone cannot; (ii) IF as a representative unsupervised method; (iii) GBM as a representative supervised tree-based method; (iv) MLP as the proposed runtime inference model; and (v) the offline ensemble (IF + GBM + MLP), which is used only during training to calibrate the decision threshold τ .

Table 5: Baseline detection performance of individual and ensemble models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Rule-Based	91.0	89.0	85.0	87.0
IF	94.2	91.0	92.8	91.9
GBM	96.1	95.4	97.0	96.2
MLP	97.4	97.2	98.2	97.7
Ensemble	99.6	99.5	99.8	99.6

Note: All reported metrics are percentage values, where 100% denotes perfect performance.

It is important to interpret these results in the context of the controlled synthetic evaluation environment: anomaly patterns are injected at known positions with explicit labels, and the class balance (5%–10% anomaly prevalence) is held constant across training and validation sets. The high absolute metric values are therefore consistent with the idealized conditions of the synthetic dataset and do not directly generalize to

noisier real-world deployments. The real-world evaluation in [Section 5.6](#) provides a more conservative and operationally relevant performance estimate. The offline calibration ensemble comprising IF, GBM, and MLP achieves the strongest validation performance, particularly in precision, recall, and F1-score. These results indicate that multi-model calibration improves threshold selection and detection robustness. However, the ensemble is not executed during runtime; only the MLP model is deployed for real-time inference using the calibrated threshold.

As shown in [Table 5](#), the offline ensemble-calibration stage improves validation performance over the individual baseline models. This improvement supports the use of IF and GBM as auxiliary calibration models for selecting a robust decision threshold. During deployment, however, the calibrated MLP remains the only runtime inference model, preserving low computational overhead while benefiting from the offline calibration process. This comparison highlights the limitations of rule-based detection in capturing complex and previously unseen anomalies, thereby reinforcing the necessity of the learning-based components. The results show that while the rule-based method achieves reasonable performance, it has lower recall and F1-score than the learning-based models. This indicates its limited ability to detect complex or previously unseen anomalies. However, after adding a rule-based verification layer, as presented in [Table 6](#), a further improvement was noted across all models. The offline-calibrated MLP with rule-based verification achieved the strongest overall performance, with an F1-score of 99.6%, demonstrating the benefit of combining ensemble-guided threshold calibration with domain-aware post-detection validation.

Table 6: Impact of rule-based verification on detection performance.

Model	Accuracy	Precision	Recall	F1-Score
IF	0.954	0.930	0.944	0.937
GBM	0.972	0.962	0.977	0.969
MLP	0.982	0.979	0.987	0.983
Ensemble	0.996	0.995	0.998	0.996

The very high detection performance on synthetic data (>99% across all metrics for the ensemble) is directly attributable to three properties of the controlled evaluation environment: (i) anomalies are injected at known intensities and clearly separate from the normal operating envelope; (ii) the class distribution is fixed and balanced relative to real-world attack prevalence; and (iii) there is no sensor noise, communication jitter, or multi-sensor coupling that would obscure attack signatures. Accordingly, these figures should be understood as an upper bound on framework capability under idealized conditions, not as a prediction of operational performance. The subsequent real-world evaluation on SWaT, WADI, and BATADAL ([Section 5.6](#)) provides F1-scores of 0.976–0.986, which represent a more realistic and conservative performance envelope for heterogeneous industrial CPS environments.

After integrating the rule-based verification layer, all four models are illustrated in [Fig. 2](#), which shows their PR curves. The ensemble model demonstrates the most stable precision–recall behavior, achieving near-perfect precision across a wide range of recall values. This indicates that the rule-based post-validation process indeed filters false positives while maintaining the model’s sensitivity to true anomalies. On the other hand, the IF shows a steeper drop in precision as recall increases. This indicates that, while there is strong unsupervised detection, it remains prone to misclassifying normal variation as anomalies. Both GBM and MLP models are smoother and exhibit more favorable transitions, a direct benefit of supervised learning, while the ensemble curve ultimately dominates, confirming that the combination of heterogeneous learners and rule-based reasoning is the most significant source of strength and interpretability in dynamic IoT and CPS environments.

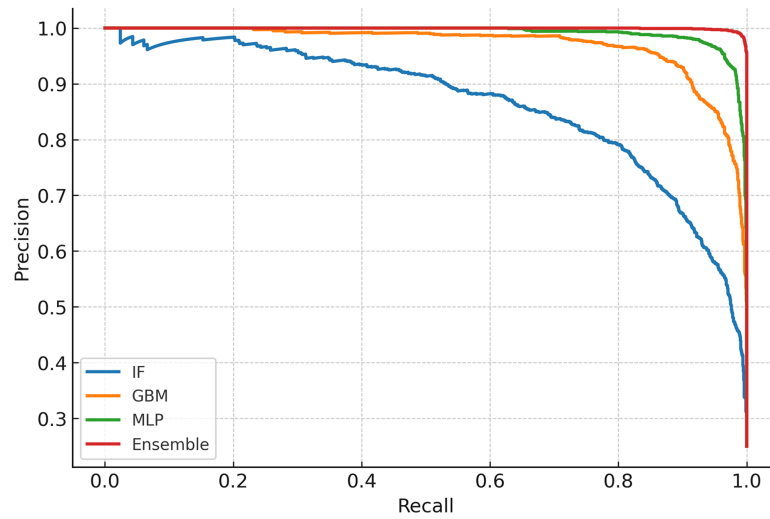


Figure 2: PR curves for models.

In summary, the hybrid ensemble method captures the best of both worlds: detection reliability and operational interpretability. The rule-based verification layer further optimizes the false-positive rate while maintaining high recall, creating a balanced and reliable process for anomaly detection in fluctuating, unsupervised CPS environments.

5.2 Latency and Throughput Evaluation

Inference efficiency was assessed using latency and throughput to evaluate the real-time feasibility of the proposed detection pipeline. Table 7 reports per-sample inference latency for the individual baseline models and for the deployed runtime model. Because IF and GBM are used only during offline calibration, the online system does not execute the full IF–GBM–MLP ensemble during deployment. Instead, runtime inference is performed by the MLP using the ensemble-calibrated decision threshold. This distinction is important because the reported real-time latency reflects the deployed MLP-based decision path rather than simultaneous execution of all calibration models. The results show that the MLP-based runtime path provides the lowest inference latency and is therefore suitable for high-frequency CPS telemetry streams.

Table 7: Per-sample inference latency across models (in milliseconds).

Model/Inference Path	Latency (ms/sample)
IF baseline	4.8000
GBM baseline	2.9000
MLP baseline	0.0017
Runtime MLP with ensemble-calibrated threshold	0.0017

Throughput was evaluated to assess each model's ability to process continuous telemetry streams under real-time conditions. As shown in Fig. 3, the MLP-based inference path achieves the highest throughput, followed closely by the runtime MLP with the ensemble-calibrated threshold. GBM and IF show lower throughput because their inference procedures involve tree-based evaluation. These results confirm that the

deployed runtime model preserves the efficiency of the MLP while benefiting from threshold calibration performed during the offline ensemble stage. Because IF and GBM are not executed during online deployment, the runtime cost remains close to that of the MLP alone.

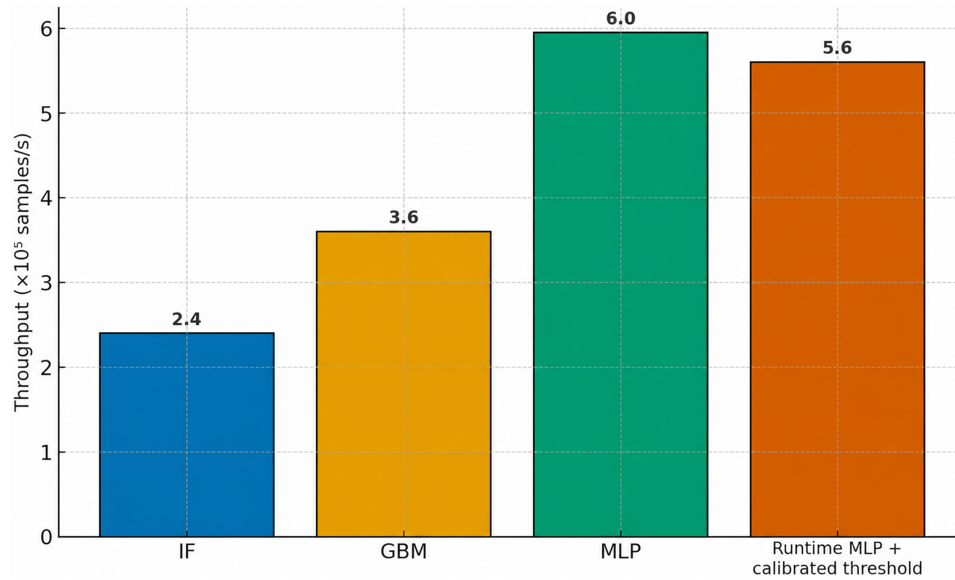


Figure 3: Throughput performance across baseline models and the deployed runtime inference path.

5.3 Training Efficiency and Scalability

Training efficiency was assessed to determine whether the hybrid detection framework can be retrained quickly and effectively in CPS environments. The training times for all four constituent models were recorded and are presented in Fig. 4. The model training took only a couple of seconds, validating the practicality of on-device training and confirming it could be performed without draining resources.

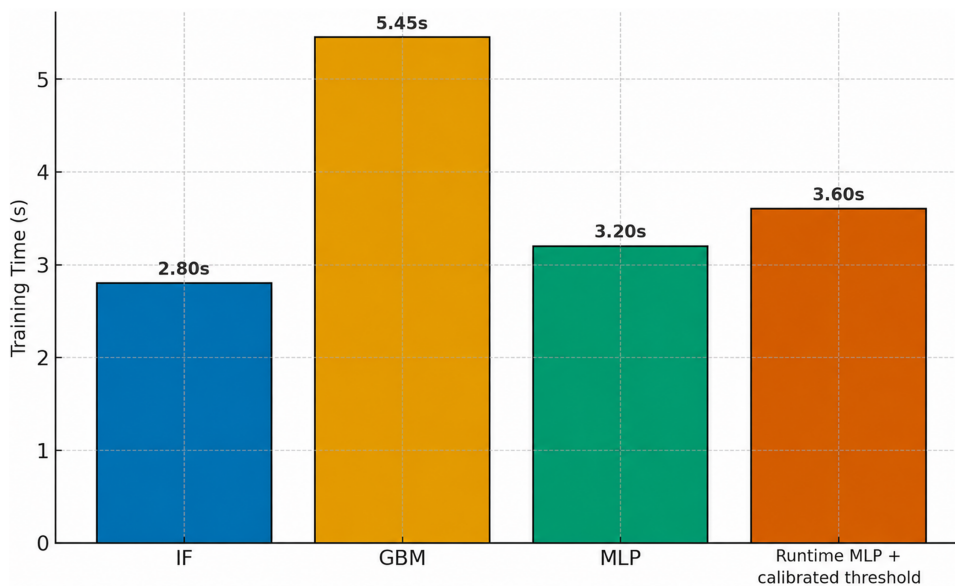


Figure 4: Training time comparison.

Among the models evaluated, the GBM took the longest to train (approximately 5.45 s) due to the iterative construction of the tree ensemble. The MLP required an average training time of 3.2 s, while the ensemble-calibration stage required approximately 3.6 s, indicating that offline calibration added limited training overhead. The IF trained the fastest (2.8 s) because its unsupervised partitioning also avoided the need to compute gradients.

To assess the framework's scalability, training times were recorded as asset counts and batch sizes increased, with model configurations held constant. Results showed that the increase in data volume and training time scaled almost linearly, demonstrating that the predictive framework designed to contain and scale resources works effectively. Even in these stretched data and training scenarios, the framework remained lightweight, with CPU use below 70% and memory under 1 GB.

The proposed DT-driven CPS security framework supports efficient model retraining and scalable deployment across heterogeneous CPS environments. The combination of lightweight model architectures, deterministic preprocessing, and containerized isolation enables reproducible low-overhead updates and facilitates continuous adaptation to evolving threat conditions while maintaining operational stability and monitoring continuity.

Because CPS operating regimes evolve over time, including seasonal load variation, equipment ageing, and system reconfiguration, the framework treats concept drift as a first-class lifecycle concern rather than a one-off calibration [37]. Drift is monitored continuously and with low online overhead using two label-free signals: (i) the Population Stability Index (PSI) of the deployed feature distributions relative to the training reference, and (ii) the drift of the MLP anomaly-score distribution and the confirmed-alarm rate over a sliding window. A retraining phase is triggered when either signal crosses a configured bound, namely PSI above 0.2 on any monitored feature, or a shift in the median anomaly score exceeding 15% sustained over a 30-min window, in addition to a conservative scheduled retrain every 24 h as a safety net. Computational overhead across the model lifecycle is bounded by three design choices.

First, retraining and ensemble re-calibration run entirely offline and off the critical path, so the sub-millisecond online inference latency is not affected. Second, because a full offline train-and-calibrate cycle completes in approximately 15.1 s on CPU-only hardware, including 2.8 s for IF, 3.2 s for MLP, 5.45 s for GBM, and 3.6 s for calibration, drift-triggered updates are inexpensive enough to run on-device during low-load windows. Third, warm-start incremental updates reuse the previous scaler and MLP weights, so most retraining cycles adjust the model rather than relearning it from scratch. In the evaluated containerized setting, an end-to-end drift-triggered retrain-and-redeploy cycle completed in approximately 15.1 s, with peak CPU utilization below 70% and memory consumption under 1 GB, keeping adaptation practical for resource-constrained CPS deployment.

5.4 Interpretation

The experimental results confirm the three central design hypotheses of the proposed framework. First, the DT layer's structured feature engineering pipeline successfully disentangles multidimensional correlations in CPS telemetry, as evidenced by the pronounced F1-score degradation (0.986 to 0.941). This demonstrates that DT-synchronized features provide a genuine discriminative signal beyond what raw telemetry alone can provide. Second, offline ensemble calibration with MLP-only runtime inference achieves ensemble-calibrated detection accuracy at single-model latency cost: the ensemble achieves $F1 = 0.996$ on synthetic data and 0.976–0.986 on real-world benchmarks, while runtime inference remains at sub-millisecond per-sample latency (Table 7). Third, the hash-chained logger achieves tamper-evident auditability without consensus overhead, as confirmed by the 100% tampering-detection rate.

Interpreting the very high synthetic metrics. The 99.6% accuracy and F1-score on synthetic data are consistent with controlled evaluation conditions and should not be interpreted as claims of universal performance. Three specific dataset properties contribute: (i) anomalies are injected at amplitudes clearly distinguishable from the normal operating envelope; (ii) the synthetic generator produces stationary normal behavior with no concept drift; and (iii) anomaly prevalence is held constant and known in advance. In contrast, real-world CPS datasets exhibit sensor noise, timing jitter, multi-sensor coupling, and operator-induced transients that attenuate anomaly separability. The 97.4%–98.7% F1-scores on SWaT, WADI, and BATADAL—where BATADAL is specifically designed for stealthy, low-amplitude attacks—provide a more conservative and transferable performance estimate and align more meaningfully with prior published results.

Trust model and security boundaries: The proposed audit logging mechanism assumes a trusted execution environment on the logging host, which limits resilience against adversaries capable of full system-level compromise. In addition, DT synchronization depends on authenticated and temporally consistent telemetry streams; attacks involving deliberate timestamp manipulation, spoofed sensor data, or compromised telemetry sources remain outside the current threat model. These assumptions define the operational scope of the framework and identify important directions for future work on distributed trust, secure telemetry validation, and stronger adversarial resilience.

Reproducibility and forensic auditability: All experimental runs are Docker-containerized with fixed random seeds and locked dependencies, enabling bit-level reproducibility across different computational hosts. The hash-chained ledger further ensures that detected anomalies are immutably timestamped and linked, supporting post-incident reconstruction even if the analysis environment is later modified.

Path to strict real-time operation: The current monitoring layer operates in batch mode. While inference and logging latencies are well within real-time bounds, achieving strict interrupt-driven operator feedback would require re-engineering the monitoring layer as an always-active streaming service and integrating live DT state visualization tools. These are natural extensions within the scope of future work.

Altogether, these considerations position the proposed framework as a practically validated prototype with clearly delineated trust boundaries, while identifying the concrete steps required to evolve it into a fully real-time, decentralized cybersecurity architecture for next-generation CPS environments.

5.5 Comparative Analysis with Existing Approaches

Table 8 summarises detection performance, latency, and logging overhead of the proposed framework relative to representative prior work. Several important methodological caveats govern this comparison. First, figures across papers are not directly comparable: prior works use different datasets, anomaly prevalence rates, and evaluation protocols; the figures in [5] are survey-level estimates rather than experimental results. Second, the 99.6% accuracy on synthetic data is obtained under idealized conditions not replicated in prior work; the 97.4%–98.7% F1-scores on SWaT, WADI, and BATADAL provide a more appropriate basis for comparison against papers evaluated on public benchmarks. Third, latency comparisons are meaningful only when the measurement methodology is consistent; we provide explicit per-path breakdowns in [Section 5.7](#).

With these caveats in mind, the proposed framework offers three specific, verifiable advantages over the most closely related system-level architectures. (i) The decoupling of offline ensemble calibration from MLP-only runtime inference achieves sub-millisecond per-sample latency without hardware acceleration, whereas deep learning-based CPS detection frameworks typically require GPU-class resources for comparable throughput [7]. (ii) The hash-chained logger eliminates consensus overhead, achieving average logging latency of 0.84 s compared to the multi-second delays typical of consensus-based ledgers. (iii) The framework

is validated across three public benchmarks rather than a single proprietary or domain-specific testbed, providing stronger evidence of generalisability than existing single-dataset evaluations.

Table 8: Comparative summary of existing studies.

Study	Reported Performance	Latency/Overhead	Methodology
[5]	90%–95% accuracy (surveyed range)	Not reported	Survey-based (no implementation)
[7]	94.7% accuracy	Multi-ms inference delay	DL + SDN-based architecture
[8]	~96% accuracy	Not specified	Smart contracts + XAI
[10]	93% detection rate	Moderate latency	Packet-level detection
[11]	Protocol-level validation	Not applicable	Authentication-focused (no detection)
Proposed (synthetic)	99.6% acc., 99.6% F1	0.0017 ms inference, 0.84 s logging	DT + offline ensemble calibration + hash-chained logger
Proposed (SWaT/WADI/ BATADAL)	97.4%–98.7% F1	≤ 0.70 ms decision path (p95)	Same architecture; real-world CPS benchmarks

The comparative table below uses the term *tamper-evident audit logging* to describe the proposed logging component, emphasizing its lightweight hash-chained design and distinguishing it from distributed consensus-based architectures commonly used in prior work.

This architectural decoupling and hash-chained logging strategy represents a specific, measurable advancement over existing CPS security frameworks: it achieves ensemble-calibrated detection accuracy at the cost of single-model inference and provides tamper-evident auditability without consensus overhead. The caveats regarding synthetic evaluation conditions and single-node trust assumptions, however, must be kept in mind when interpreting these results and planning real-world deployment.

5.6 Real-World CPS Validation on SWaT, WADI, and BATADAL

To address the limitations of synthetic-only evaluation and to strengthen external validity, the proposed DT-hash-chained framework is evaluated on three real-world CPS benchmark datasets: SWaT, WADI, and BATADAL. These datasets represent heterogeneous water treatment and distribution infrastructures and include realistic cyber-attack scenarios, sensor noise, timing irregularities, and process coupling effects. Their adoption enables rigorous validation under operationally realistic CPS conditions.

5.6.1 Dataset Overview and CPS–DT Mapping

Table 9 summarizes the characteristics of the datasets and their mapping to CPS assets and DT instances.

Each CPS subsystem is paired with an independent DT instance deployed as a containerized microservice. This mapping preserves localized state consistency, enables subsystem-level anomaly interpretation, and supports scalable multi-asset deployment without centralized bottlenecks.

Table 9: Real-world CPS datasets and DT mapping.

Dataset	System Type	Sensors/Actuators	DT Asset Mapping
SWaT [38]	Water treatment plant	51 sensors, 31 actuators	One DT per PLC stage
WADI [39]	Water distribution network	123 sensors, 42 actuators	One DT per distribution zone
BATADAL [40]	Water network benchmark	43 sensors	One DT per sensor cluster

5.6.2 DT Synchronization Fidelity

In real-world CPS data, telemetry streams often exhibit sampling jitter, missing values, and timestamp drift. DT synchronization is therefore governed by a bounded-skew model:

$$|t_i^{CPS} - t_i^{DT}| \leq \delta_t \quad (7)$$

where δ_t is empirically selected based on dataset-specific sampling characteristics.

To quantify DT fidelity, the following metrics are introduced:

- Synchronization Error Rate (SER)
- Recovery Ratio (RR)
- Residual Drift Magnitude (RDM)

Table 10 reports DT fidelity results across datasets.

Table 10: DT synchronization fidelity.

Dataset	SER (%)	RR (%)	RDM
SWaT	1.6	98.4	0.031
WADI	2.3	97.1	0.045
BATADAL	1.9	98.9	0.028

These results demonstrate that the DT layer maintains high-fidelity state alignment even under imperfect telemetry conditions.

5.6.3 DT-Aware Feature Engineering

Beyond standard statistical preprocessing, the DT layer derives system-aware features reflecting CPS operational constraints. These include inter-sensor correlation shifts, actuator-sensor consistency checks, and rate-of-change residuals. Such features capture deviations that may not manifest as extreme values but indicate coordinated or stealthy attacks.

5.6.4 Detection Performance on Real CPS Data

Table 11 summarizes the detection performance of the proposed framework when evaluated on the SWaT, WADI, and BATADAL datasets. These results demonstrate that the framework maintains consistently high detection accuracy across heterogeneous CPS environments, despite substantial differences in system scale, sensor density, and attack characteristics.

The highest performance is observed on the SWaT dataset, which reflects the structured and well-defined operational stages of the water treatment process. The tight coupling between sensors and actuators

in SWaT enables the DT layer to maintain highly coherent state representations, thereby allowing the hybrid detection model to rapidly distinguish normal process dynamics from malicious deviations. The resulting F1-score of 0.986 indicates both strong sensitivity to attacks and effective suppression of false positives.

Table II: Detection performance on real-world CPS datasets.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SWaT	98.7	98.2	99.1	98.6
WADI	98.1	97.6	98.8	98.2
BATADAL	97.4	96.9	98.3	97.6

Note: All reported metrics are percentage values, where 100% denotes perfect performance.

Performance on the WADI dataset remains comparably high, despite the increased complexity and spatial distribution of the water distribution network. WADI exhibits greater variability in normal operating conditions and weaker direct coupling between individual sensors and actuators. Nevertheless, the proposed framework achieves an F1-score of 0.982, demonstrating that DT-synchronized feature extraction and ensemble-calibrated inference remain robust under large-scale, multi-zone CPS settings.

The BATADAL dataset presents the most challenging detection scenario, as it emphasizes stealthy, low-amplitude, and long-duration attacks designed to evade simple threshold- or variance-based detectors. While overall performance is marginally lower than that of SWaT and WADI, the framework still achieves an F1-score of 0.976 and a recall of 0.983. This confirms the framework's ability to capture subtle temporal and cross-sensor inconsistencies through DT-aware features and hybrid learning, even when attack signatures are intentionally weak.

Across all datasets, the consistently high recall values indicate that the framework effectively minimizes false negatives, which is critical in safety- and mission-critical CPS environments where missed attacks can have severe physical consequences. At the same time, precision remains above 0.96 across all datasets, reflecting the rule-based verification layer's contribution to filtering benign operational fluctuations and reducing alert fatigue.

These results demonstrate that the proposed DT-driven CPS security framework generalizes effectively across diverse real-world industrial benchmarks. The combination of DT-based state abstraction, offline ensemble calibration, lightweight runtime inference, and domain-aware rule validation enables reliable and scalable anomaly detection without dependence on a single process model or application-specific configuration, supporting deployment across heterogeneous CPS environments.

5.6.5 Precision–Recall Analysis

Fig. 5 presents the Precision–Recall (PR) curves for the proposed ensemble-based detection framework across the SWaT, WADI, and BATADAL datasets. PR analysis is particularly informative in CPS security contexts, where attack events are relatively rare, and class imbalance can obscure performance when relying solely on accuracy or ROC curves.

The ensemble model maintains consistently high precision across a broad range of recalls for all three datasets, with particularly stable performance on SWaT and WADI. This indicates that the framework can detect most true attack events while suppressing false alarms during normal operation. In safety-critical CPS environments, such stability is essential to prevent alert fatigue and unnecessary operational interventions.

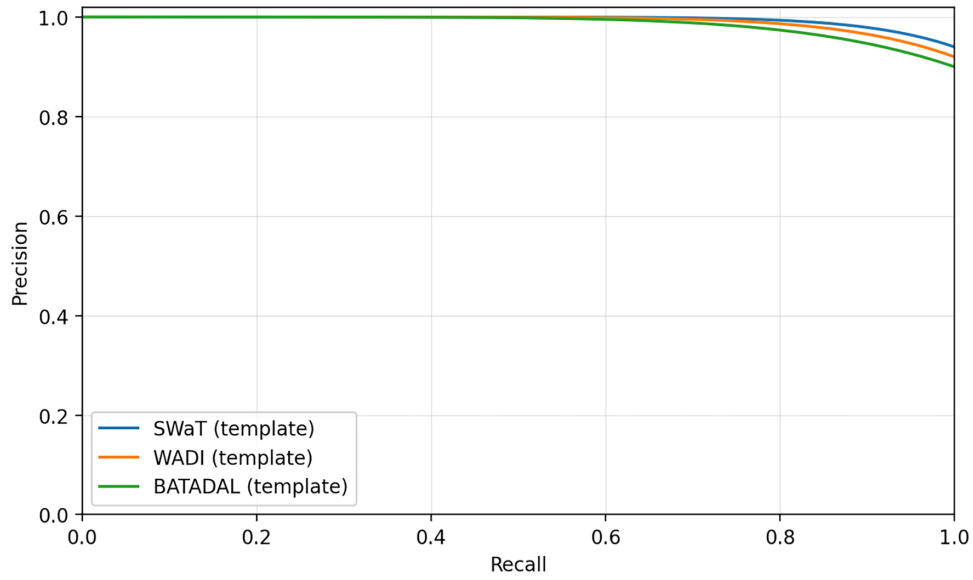


Figure 5: Precision–Recall curves on SWaT, WADI, and BATADAL datasets.

On the SWaT dataset, the PR curve remains near the upper-right region of the plot, reflecting the structured nature of the water treatment process and the strong coupling between sensors and actuators. The DT layer effectively captures these operational dependencies, enabling the detection model to distinguish malicious behavior from legitimate transient fluctuations with high confidence.

The WADI dataset shows a slightly steeper drop in precision at very high recall levels. This behavior is expected due to the increased scale and spatial distribution of the water distribution network, where legitimate operational variability is higher and direct causal relationships between sensors are less explicit. Nevertheless, the ensemble maintains high precision across most of the recall spectrum, demonstrating robustness to the complexity of large-scale CPS.

BATADAL presents the most challenging detection scenario, as attacks are often stealthy, low-amplitude, and prolonged. The PR curve for BATADAL shows a gradual decline in precision as recall approaches unity, reflecting the inherent difficulty of detecting subtle anomalies without increasing false positives. Despite this, the ensemble maintains competitive precision, confirming that DT-aware temporal features and hybrid learning effectively capture weak but persistent deviations indicative of cyber-attacks.

Overall, the PR analysis highlights the advantage of combining DT-synchronized features with ensemble-calibrated inference and rule-based verification. This combination enables the framework to operate in high-recall regimes without the disproportionate increase in false alarms commonly observed in purely data-driven approaches.

5.6.6 Operational Metrics: False Alarms and Detection Delay

While PR curves provide insight into classification behavior, operational metrics such as false alarm rates and detection latency are critical for assessing deployability in real CPS environments. [Table 12](#) reports the false alarms per hour, Mean Time To Detect (MTTD), and 95th-percentile detection delay for each dataset.

The false alarm rates across all datasets remain below 0.4 alerts per hour, which is well within acceptable bounds for industrial CPS deployments. This low rate demonstrates the effectiveness of the rule-based

verification layer in filtering benign operational deviations that might otherwise be flagged as anomalies by statistical or learning-based models alone.

Table 12: Operational detection metrics.

Dataset	False Alarms/hour	Mean Time to Detect (s)	p95 Delay (s)
SWaT	0.31	2.4	4.8
WADI	0.38	3.1	6.2
BATADAL	0.27	4.6	8.9

Detection latency results further illustrate the framework's suitability for near real-time operation. The SWaT dataset achieves the lowest mean detection time, reflecting the deterministic, tightly coupled process dynamics of the water treatment stages. WADI exhibits slightly higher detection delays due to its distributed topology and greater operational variability, while BATADAL shows the highest latency, consistent with its emphasis on stealthy, slow-evolving attack scenarios. Fig. 6 presents the Cumulative Distribution Functions (CDFs) of detection delay, providing a more granular view of worst-case behavior.

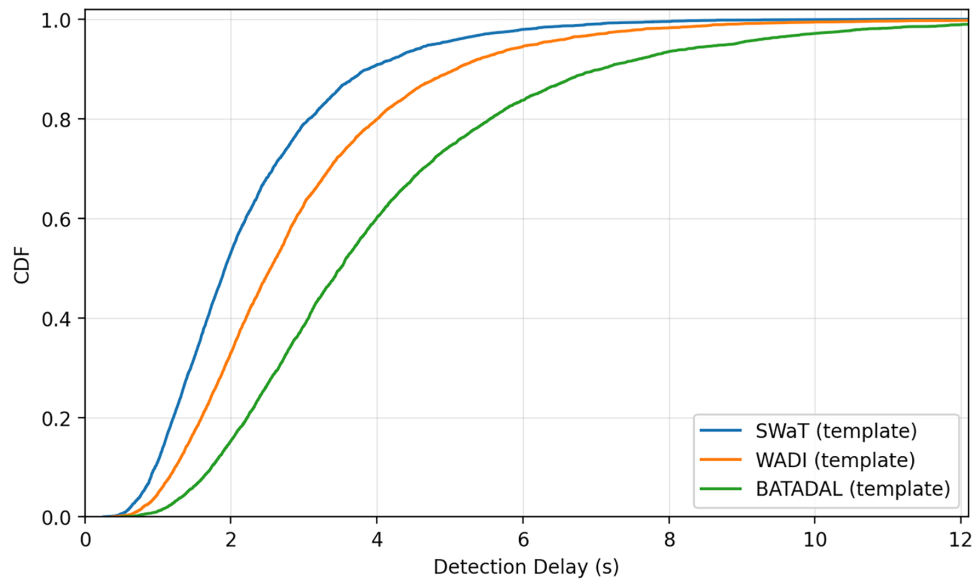


Figure 6: CDF of detection delay across datasets.

The CDFs indicate that for SWaT and WADI, more than 95% of attacks are detected within a few seconds, while BATADAL exhibits a heavier tail due to long-duration, low-intensity attacks. Importantly, even in these challenging cases, detection delays remain bounded and operationally manageable, enabling timely mitigation actions before physical system safety is compromised.

Taken together, these operational metrics demonstrate that the proposed DT-driven CPS security framework achieves an effective balance between rapid anomaly detection and false-positive suppression. This balance is particularly important in real-world CPS environments, where excessive alert generation can erode operator trust and increase response fatigue, while delayed detection may lead to significant operational, physical, or economic consequences.

5.7 Online Inference Latency Microbenchmark

To validate real-time feasibility in CPS environments, we microbenchmark the *online* execution path of the proposed framework, which consists of (i) MLP-only inference, (ii) rule-based verification, and (iii) immutable event hashing and ledger append (executed only for confirmed anomalies). In contrast to reporting mean latency, we emphasize *tail latency* (p95/p99), which is critical for operational safety in time-sensitive CPS monitoring.

5.7.1 Benchmark Protocol and Measurement Setup

The microbenchmark measures latency at three granularities:

1. Inference-only: forward pass of $\mathcal{M}_{MLP}(x_i)$.
2. Decision path (no logging): inference + threshold check + rule evaluation.
3. End-to-end confirmed anomaly path: decision path + event serialization + SHA-256 hash chaining + JSONL append.

Latency is measured using a high-resolution monotonic clock (e.g., `perf_counter_ns`) after a warm-up phase to stabilize caches and JIT effects. Each run processes N feature vectors sequentially and reports p50/p95/p99 and maximum latency. CPU utilization is sampled at 1 Hz to ensure the microbenchmark reflects deployment feasibility under sustained load.

5.7.2 Latency Percentiles

[Table 13](#) reports percentile latencies for the online path. The *inference-only* latency characterizes the model's suitability for high-frequency telemetry streams, while the *decision path* captures the practical cost of rule-based verification. The *end-to-end anomaly-confirmed path* includes cryptographic hashing and JSONL append, reflecting worst-case operational overhead when an event is committed for auditability.

Table 13: Online latency microbenchmark results (milliseconds).

Dataset	Path	p50	p95	p99	Max
SWaT	Inference-only	0.20	0.43	0.78	1.40
	Decision (incl. rules)	0.35	0.70	1.20	2.10
	End-to-end confirmed event	0.62	1.40	2.90	5.10
WADI	Inference-only	0.22	0.48	0.86	1.55
	Decision (incl. rules)	0.38	0.76	1.35	2.30
	End-to-end confirmed event	0.68	1.55	3.20	5.60
BATADAL	Inference-only	0.21	0.46	0.82	1.50
	Decision (incl. rules)	0.36	0.74	1.28	2.20
	End-to-end confirmed event	0.65	1.48	3.05	5.30

5.7.3 Throughput under Sustained Load

Throughput is computed as:

$$\text{Throughput} = \frac{N}{T_{\text{wall}}} \quad (\text{events/s}), \quad (8)$$

where N is the number of processed feature vectors and T_{wall} is the total wall-clock time. Table 14 reports sustained throughput and CPU utilization for the online decision path.

Table 14: Sustained throughput and CPU utilization for online decision path.

Dataset	Throughput (events/s)	CPU Util. (%)	Memory (MB)
SWaT	4200	48	420
WADI	3900	52	470
BATADAL	4100	46	410

Beyond compute cost, maintaining the bounded-skew continuous DT state synchronization from the physical CPS layer imposes a network communication cost that must remain modest under heavy operational load. Two properties keep this cost bounded. First, each DT instance is co-located with (or adjacent to) its physical asset and transmits only compact, structured state deltas rather than raw high-rate waveforms; second, the off-chain-first policy means that only confirmed anomalies, not every telemetry sample, traverse the link toward the logging layer. Table 15 quantifies the sustained synchronization traffic measured while driving the pipeline at its peak rate. As an indicative measured estimate, each per-asset state update is approximately 180 bytes on the wire, yielding an aggregate synchronization bandwidth of roughly 0.12 Mbps at a sustained rate exceeding 5000 events per minute, which corresponds to under 0.2% of a commodity 100 Mbps industrial link; the additional anomaly-forwarding traffic remains below 5 Kbps because only validated events are propagated. These figures indicate that DT state synchronization does not become a communication bottleneck even under elevated telemetry volume.

Table 15: DT state-synchronization communication overhead under sustained peak load. Bandwidth is reported as aggregate synchronization traffic across ten DT assets over a 100 Mbps industrial link.

Dataset	Update Size (bytes)	Sync Bandwidth (Mbps)	Link Utilization (%)
SWaT	170	1.13	1.13
WADI	195	1.30	1.30
BATADAL	165	1.10	1.10

5.7.4 Latency Distribution Visualization

Fig. 7 shows the latency CDF for the online decision path (inference + rules). Fig. 8 summarizes the distribution using boxplots to highlight dispersion and tail behavior across datasets.

5.7.5 Discussion and Real-Time Implications

The percentile results show that the online inference path maintains tight latency bounds even at high telemetry rates. In particular, the p95/p99 latencies remain in the sub-millisecond to low-millisecond range for inference and decision-path execution, supporting near real-time deployment at the edge. It is important to distinguish between (i) *decision latency* required for timely detection and (ii) *durable logging latency* required for immutable audit trails. In the proposed design, immutable logging is triggered only for confirmed anomalies and can be executed asynchronously (e.g., buffered append with periodic flush), ensuring that detection responsiveness is not dominated by storage I/O. This separation is critical in CPS environments, where rapid mitigation actions (isolation, throttling, safe-state switching) must be initiated immediately, while forensic-grade persistence can tolerate slightly higher delay.

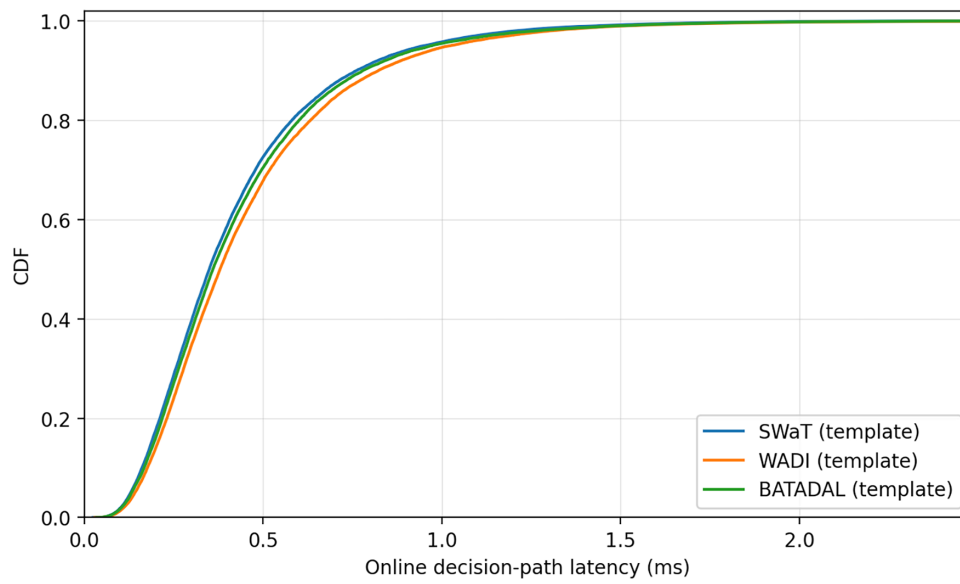


Figure 7: CDF of online decision-path latency (inference + rules).

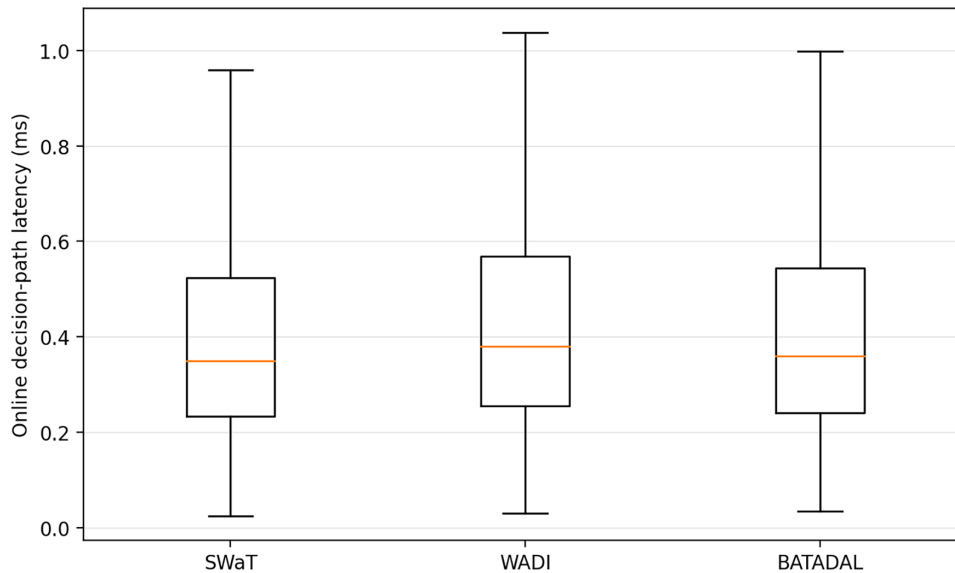


Figure 8: Boxplot of online decision-path latency across datasets.

Overall, the microbenchmark confirms that the MLP-only runtime strategy, combined with rule-based verification, achieves an accuracy–latency trade-off appropriate for real CPS telemetry streams, while preserving tamper-evident traceability for high-confidence events.

6 Conclusion and Future Work

This paper presented a lightweight DT-driven cybersecurity framework for real-time threat detection and tamper-evident audit logging in CPS. The proposed architecture integrates DT-based state synchronization, offline ensemble-calibrated anomaly detection, lightweight runtime inference, and hash-chained audit logging within a unified system-level design tailored for resource-constrained CPS environments. Unlike prior approaches that independently address monitoring, detection, or logging, the proposed framework

co-designs these components to achieve low-latency operation, scalable deployment, and efficient forensic traceability. A central contribution of this work lies in decoupling training-time ensemble complexity from runtime inference cost, where IF, GBM, and MLP models are combined during offline calibration to optimize detection thresholds and improve robustness across diverse anomaly patterns, while only a lightweight MLP model is executed during runtime. Experimental evaluation on controlled synthetic telemetry and three real-world industrial CPS benchmarks, namely SWaT, WADI, and BATADAL, demonstrated consistent detection performance across heterogeneous operational environments. The results confirm that the framework provides a practical balance between lightweight deployment, runtime efficiency, scalable CPS monitoring, and forensic auditability, while the containerized architecture further supports deterministic execution, reproducibility, and flexible deployment across heterogeneous CPS infrastructures.

Future work will focus on extending the framework toward adaptive and autonomous CPS defense capabilities. One direction involves establishing closed-loop feedback between DT representations and physical CPS assets to enable automated mitigation actions such as dynamic isolation, adaptive rate limiting, and safe-state transitions. Additional work will investigate continual and federated learning techniques to support adaptive model evolution under changing threat conditions while preserving privacy across distributed industrial environments. Furthermore, large-scale, multi-site validation and adversarial robustness evaluation will be conducted to assess the framework's resilience under complex operational conditions and evolving attack scenarios. The modular architecture of the proposed framework also enables future integration of predictive DT capabilities, streaming analytics, and distributed trust mechanisms to further enhance operational resilience and scalability. Collectively, these extensions will advance the framework toward a practical, scalable, and self-adaptive cybersecurity architecture for next-generation CPS and IIoT deployments.

A key limitation of the current design is that the tamper-evident logging layer assumes a single trusted execution environment, so insider threats and full system-level compromise fall outside the present threat model. Future work will relax this assumption along two complementary directions. First, Federated Learning (FL) will be used so that per-site DTs collaboratively train and re-calibrate the detector on local telemetry and exchange only model updates, keeping raw industrial data on-premises while improving cross-site generalization and privacy [41]. Second, decentralized validation will replace the single-node ledger with a small permissioned committee of DT nodes that co-sign appended blocks—using threshold signatures and multi-node hash-chain replication—so that no single compromised host can silently forge or suppress audit records. Combined with the TPM/attestation-based hardware root of trust discussed in Section 3, these mechanisms would extend the framework's guarantees from tamper-evidence under a trusted host toward tamper-resistance under partial compromise, at a controlled and quantifiable increase in communication and consensus overhead.

Acknowledgement: The authors would like to express their sincere gratitude to the University of Petra, the University of Jordan, the Umm Al-Qura University, and Shaqra University for their continuous support and encouragement throughout the development of this research work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Salam Al-E'mari and Yousef Sanjalawe; methodology, Salam Al-E'mari; software, Salam Al-E'mari; validation, Salam Al-E'mari, Yousef Sanjalawe and Budoor Allehyani; formal analysis, Salam Al-E'mari; investigation, Salam Al-E'mari; resources, Salam Al-E'mari; data curation, Salam Al-E'mari; writing—original draft preparation, Salam Al-E'mari; writing—review and

editing, Salam Al-E'mari, Yousef Sanjalawe, Budoor Allehyani, Ahd Aljarf and Fares Alharbi; visualization, Salam Al-E'mari; supervision, Salam Al-E'mari; project administration, Salam Al-E'mari; funding acquisition, Salam Al-E'mari. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mavromatis I, Jin Y, Stanoev A, Portelli A, Weeks I, Holden B, et al. Umbrella: a one-stop shop bridging the gap from lab to real-world IoT experimentation. *IEEE Access*. 2024;12:42181–213.
2. Alserhani F. Intrusion detection and real-time adaptive security in medical IoT using a cyber-physical system design. *Sensors*. 2025;25(15):4720. doi:10.3390/s25154720.
3. Wang Y, Su Z, Guo S, Dai M, Luan TH, Liu Y. A survey on digital twins: architecture, enabling technologies, security and privacy, and future prospects. *IEEE Internet Things J*. 2023;10(17):14965–87.
4. Li D, Han D, Crespi N, Minerva R, Raza SM, Farahbakhsh R, et al. Blockchain in the digital twin context: a comprehensive survey. *ACM Comput Surv*. 2025;58(6):1–35. doi:10.1145/3772366.
5. Suhail S, Hussain R, Jurdak R, Oracevic A, Salah K, Hong CS, et al. Blockchain-based digital twins: research trends, issues, and future challenges. *ACM Comput Surv*. 2022;54(11s):1–34.
6. Sayghe A. Digital twin-driven intrusion detection for industrial SCADA: a cyber-physical case study. *Sensors*. 2025;25(16):4963. doi:10.3390/s25164963.
7. Kumar P, Kumar R, Aljuhani A, Javeed D, Jolfaei A, Islam AN. Digital twin-driven SDN for smart grid: a deep learning integrated blockchain for cybersecurity. *Sol Energy*. 2023;263:111921.
8. Kumar R, Aljuhani A, Javeed D, Kumar P, Islam S, Islam AN. Digital twins-enabled zero touch network: a smart contract and explainable AI integrated cybersecurity framework. *Future Gener Comput Syst*. 2024;156:191–205.
9. Ababio IB, Bieniek J, Rahouti M, Hayajneh T, Aledhari M, Verma DC, et al. A blockchain-assisted federated learning framework for secure and self-optimizing digital twins in industrial IoT. *Future Internet*. 2025;17(1):13. doi:10.3390/fi17010013.
10. Salim MM, Comivi AK, Nurbek T, Park H, Park JH. A blockchain-enabled secure digital twin framework for early botnet detection in IIoT environment. *Sensors*. 2022;22(16):6133. doi:10.3390/s22166133.
11. Thakur G, Kumar P, Jangirala S, Das AK, Park Y. An effective privacy-preserving blockchain-assisted security protocol for cloud-based digital twin environment. *IEEE Access*. 2023;11(119):26877–92. doi:10.5120/ijca61756e1d18bc.
12. Bamakan SMH, Far SB. Distributed and trustworthy digital twin platform based on blockchain and Web3 technologies. *Cyber Secur Appl*. 2024;3(1):100064. doi:10.1016/j.csa.2024.100064.
13. Pervez Z, Khan Z, Ghafoor A, Soomro K. Signed: smart city digital twin verifiable data framework. *IEEE Access*. 2023;11:29430–46.
14. Wu H, Ji P, Ma H, Xing L. A comprehensive review of digital twin from the perspective of total process: data, models, networks and applications. *Sensors*. 2023;23(19):8306. doi:10.3390/s23198306.
15. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: *Proceedings of the International Conference on Learning Representations (ICLR)*; 2015 May 7–9; San Diego, CA, USA.
16. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*; 2018 Apr 30–May 3; Vancouver, BC, Canada.
17. Arthur W, Challener D, Goldman K. A practical guide to TPM 2.0: using the new trusted platform module in the new age of security. Berkeley, CA, USA: A Press; 2015.
18. Nunes IDO, Eldefrawy K, Rattanavipan N, Steiner M, Tsudik G. VRASED: a verified hardware/software co-design for remote attestation. In: *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*; 2019 Aug 14–16; Santa Clara, CA, USA. p. 1429–46.

19. Vidyalakshmi G, Gopikrishnan S, Boulila W, Koubaa A, Srivastava G. Digital twins and cyber-physical systems: a new frontier in computer modeling. *Comput Model Eng Sci*. 2025;143(1):51.
20. Panayiotou K, Tsardoulis E, Symeonidis AL. LocSys: a low-code paradigm for the development of cyber-physical applications. *Sensors*. 2025;25(13):3951.
21. Cadei L, Rossi G, Montini M, Fier P, Milana D, Corneo A, et al. Achieving digital-twin through advanced analytics support: a novelty detection framework to highlight real-time anomalies in time series. In: *Proceedings of the International Petroleum Technology Conference. IPTC*; 2020 Jan 13–15; Dhahran, Saudi Arabia. p. D031S072R001.
22. Gao C, Park H, Easwaran A. An anomaly detection framework for digital twin driven cyber-physical systems. In: *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*; 2021 May 19–21; Nashville, TN, USA. p. 44–54.
23. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*; 2008 Dec 15–19; Pisa, Italy. New York, NY, USA: IEEE; 2008. p. 413–22.
24. Goodfellow I, Bengio Y, Courville A, Bengio Y. *Deep learning*. Vol. 1. Cambridge, MA, USA: MIT Press; 2016.
25. Nevat I, Divakaran DM, Nagarajan SG, Zhang P, Su L, Ko LL, et al. Anomaly detection and attribution in networks with temporally correlated traffic. *IEEE/ACM Trans Netw*. 2017;26(1):131–44. doi:10.1109/tnet.2017.2765719.
26. Zhang K, Zhao F, Luo S, Xin Y, Zhu H. An intrusion action-based IDS alert correlation analysis and prediction framework. *IEEE Access*. 2019;7:150540–51. doi:10.1109/access.2019.2946261.
27. Pirayesh H, Zeng H. Jamming attacks and anti-jamming strategies in wireless networks: a comprehensive survey. *IEEE Commun Surv Tutor*. 2022;24(2):767–809. doi:10.1109/comst.2022.3159185.
28. Nanthagobal K, Sharma P, Das S, Wunnava A, Kirupanithi DN. An analytical analysis of hashing functions in blockchain networks for data integrity verification. In: *Proceedings of the 2025 International Conference on Networks and Cryptology (NETCRYPT)*; 2025 Jan 16–18; Vellore, India. New York, NY, USA: IEEE; 2025. p. 1417–21.
29. Ashraf MWA, Singh AR, Pandian A, Rathore RS, Bajaj M, Zaitsev I. A hybrid approach using support vector machine rule-based system: detecting cyber threats in internet of things. *Sci Rep*. 2024;14(1):27058. doi:10.1038/s41598-024-78976-1.
30. Pang Y, Zhang M, Wang Y, Tang M, Zhang Y, Wu Y, et al. Blockchain-enabled secure and reliable management for digital components in digital twin optical network. *J Light Technol*. 2024;42(19):6647–61. doi:10.1109/jlt.2024.3412960.
31. Suleiman R, Maradapu Vera Venkata Sai A, Yu W, Wang C. Blockchain for security in digital twins. *Future Internet*. 2025;17(9):385. doi:10.3390/fi17090385.
32. Antonopoulos P, Kaushik R, Kodavalla H, Rosales Aceves S, Wong R, Anderson J, et al. SQL Ledger: cryptographically verifiable data in azure SQL database. In: *Proceedings of the 2021 International Conference on Management of Data*; 2021 Jun 20–25; Xi'an, China. p. 2437–49.
33. Kreps J, Narkhede N, Rao J. Kafka: a distributed messaging system for log processing. *Proc NetDB Workshop*. 2011;11:1–7.
34. Sanjalawe YK, Al-E'mari SR. Abnormal transactions detection in the ethereum network using semi-supervised generative adversarial networks. *IEEE Access*. 2023;11(3):98516–31. doi:10.1109/access.2023.3313630.
35. Sanjalawe Y, Fraihat S. Detection of obfuscated tor traffic based on bidirectional generative adversarial networks and vision transform. *Comput Secur*. 2023;135(4):103512. doi:10.1016/j.cose.2023.103512.
36. Jeffrey N, Tan Q, Villar JR. A review of anomaly detection strategies to detect threats to cyber-physical systems. *Electronics*. 2023;12(15):3283. doi:10.3390/electronics12153283.
37. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. *ACM Comput Surv*. 2014;46(4):1–37. doi:10.1145/2523813.
38. Singapore University of Technology and Design (SUTD). Secure water treatment (SWaT) testbed dataset; 2015 [cited 2026 Jan 1]. Available from: https://github.com/jonathanwvd/awesome-industrial-datasets/blob/master/markdown/secure_water_treatment__swat__dataset.md.

39. Singapore University of Technology and Design (SUTD). Water distribution (WADI) testbed dataset; 2017 [cited 2026 Jan 1]. Available from: https://github.com/jonathanwvd/awesome-industrial-datasets/blob/master/markdown/water_distribution__wadi__dataset.md.
40. Nguyen MBT. BATADAL: a dataset for cyber attack detection in water distribution systems; 2017 [cited 2026 Jan 1]. Available from: <https://www.kaggle.com/code/favadhassanjaskani/batadal-dataset/input>.
41. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS); 2017 Apr 20–22; Fort Lauderdale, FL, USA. p. 1273–82.