Experimental Digital Twin Framework for Cybersecurity Risk Analysis

Nathanaël Stemmel[1] **Manuel Munier**[1] Ali Larbi[2]

[1]LIUPPA, Université de Pau et des Pays de l'Adour, E2S UPPA

[2]ETIS, CY Cergy Paris Université

MEDES 2026 @ Coruña, Spain





Context: the Digital Twin is everywhere

- Aggregates real-world data for **monitoring**, **predictive analytics** and **control** of a system.
- Fast-growing literature and deployments.

Use cases

- **Smart cities**: traffic, energy, emergency response
- **Industry 4.0**: predictive maintenance, quality control
- **Connected agriculture**: soil sensors, drones, irrigation

The other side of the coin

- What makes the DT attractive (aggregation, connectivity, AI) is also its **attack surface**
- A DT is only as reliable as **its data**

Problem and contributions

Problem

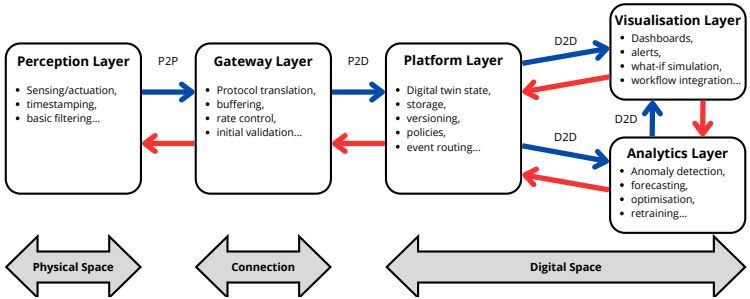
The lack of **reproducible experimental environments** hinders systematic evaluation and research on DT cybersecurity.

Three contributions:

1. A **minimal yet realistic** DT pipeline: sensor data, data governance, lightweight AI.
2. Simulation of **representative cyber threats** and evaluation of countermeasures (governance + anomaly detection).
3. A **reproducible, pedagogical** baseline for the community.

Digital twin architecture

- Three spaces: **physical** (sensors, actuators), **digital** (models, analytics), and **connection** layer (P2P, P2D, D2D).
- Layered stack: perception, gateway, platform, analytics/ML, visualisation and control.



Fragility of data governance

- A DT is only as reliable as **the data it gathers**.
- Data quality deficits:
  - **Completeness**: missing sensors, lost packets → blind spots
  - **Consistency**: schema drift, unit errors (Celsius/Kelvin), out-of-order events
  - **Timeliness**: outdated data driving real actions
  - **Provenance**: insufficient metadata → trust gaps, leaks

Common DT threats

Model poisoning · Eavesdropping and leakage · Replay · MITM tampering · Impersonation / Sybil

Countermeasures: classical/statistical vs. AI-driven

	Classical / Statistical	AI-driven
Examples	Secure boot, role-based access; rule/threshold alarms	Supervised / unsupervised anomaly detection
Strengths	Explainable, low footprint, certifiable	Complex patterns, adapts to drift, effective against unknown attacks
Limits	Weak to new attacks, slow poisoning	Requires data, false positives, footprint

Chosen strategy

Complementary roles → **hybrid** approach: statistical and metadata checks first, AI for subtle and unknown attacks.

Threats simulated in our study

Threat	Vector / layer	Effect on the DT	Implementation
Replay injection	Network, P2D	Outdated but valid readings	Re-publish captured MQTT payloads, old timestamps, higher rate
MITM tampering	Edge, P2P/P2D	Subtle sensor-value changes, unsafe actuation	Transparent broker/bridge modifying fields
Desynchronisation	Control plane	Twin lags the plant, inconsistent state	Random message drops/delays, clock perturbation

- Choices guided by two criteria: **field realism** and **lab reproducibility**.

Detection I: data quality and metadata

Data quality (DQ) checks

- **Validity**: plausible bounds, correct units
- **Timeliness**: on-time arrival
- **Completeness**: no missing fields or gaps
- **Drift**: same values, same conditions?

Rich metadata (SSN ontology)

Source, calibration, units, confidence → proven integrity, policy applied **before** the twin state.

```
/topic_used {"name": "sensor_name",
  "results": [{"label": "temp",
               "value": 20.58,
               "unit": "Celsius"}],
  "collections": ["building_A_office"],
  "featureOfInterest":
    "building_A_office_temperature"}
```

Simple checks (unit, collection, feature of interest) = immediate DQ gain.

Detection II: machine learning

- Thresholds cannot see complex patterns: AI adds recognition of **subtle, unknown attacks** (seasonal fluctuations, slow drift).
- **Unsupervised anomaly detection** (Isolation Forest, feature-based): small models, raw data kept local.

Model lifecycle

Collection → preprocessing → feature extraction → selection and training → real-time detection → evaluation → deployment

- **Gradual re-training**: the DT evolves without stopping production.

Experimental setup

Deployed stack (Docker Compose, single host)

- **Eclipse Ditto**: twin models and policies
- **MQTT broker** (Mosquitto / EMQX): telemetry
- **Python gateway**: metadata validation, quality-feature computation
- **InfluxDB**: raw time series + anomalies (timestamp, measurement, severity)

Loop: watchdog 30 s → score of the last 2 minutes → anomalies in the

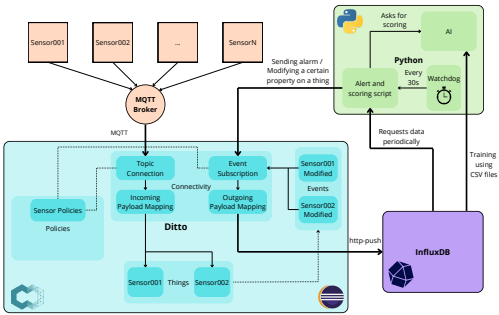


Figure: Our end-to-end installation

Results: what the AI detected

- The model did **most of the heavy lifting**: replay, MITM tampering and desynchronisation were all flagged (bad scores → quarantine).
- Even threats that statistical methods should cover (timeliness, consistency) were captured by the AI.

Observed trade-off

Detection sensitivity **vs** false-positive rate: in a real scenario, more intensive training is needed so that no correct data is ever flagged.

- Automatic re-training is not implemented, but is essential at scale (drift, seasonality).

Discussion: how much security?

- The security level of a DT depends on:
  - the **likelihood** of being attacked,
  - the **sensitivity** of the information,
  - the **power of action** on reality.
- Critical DT (sensitive data + dangerous actuation) $\rightarrow$ rigorous AI, well trained and closely monitored.
- **Combined plan**: simple, effective statistical and metadata checks relieve the AI without giving anything up.

Beyond traditional network protection

Data integrity, provenance, synchronisation and trust across the entire DT ecosystem.

Conclusion and future work

What we deliver

A **compact, modular and reproducible** DT application: a simple yet representative architecture, end-to-end simulated threats, interchangeable components (detector, database. . .).

Future work:

- Deeper **metadata** exploitation (context, environmental signatures)
- More advanced **AI models** for dynamic operations
- **Systematic quantitative analysis** of attack datasets
- **Federated learning**: limit poisoning, preserve privacy across distributed DTs

How: Launch of the **Twin-VIC** project (CyTech / UPPA) with 1 PhD thesis

Thank you!

Questions?

Experimental Digital Twin Framework for Cybersecurity Risk Analysis

▷ *Reproducible experimental setup — full stack under Docker Compose* ◁

Nathanaël Stemmel, **Manuel Munier**, Ali Larbi

LIUPPA, Université de Pau et des Pays de l'Adour, E2S UPPA
ETIS, CY Cergy Paris Université

MEDES 2026 @ Coruña, Spain