Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Automated cross-layer metamodeling for secure digital twins: attack detectability by construction

Ferdaws Ben Naceur
University of Sousse
Mahmoud El Amri
mahmoudelamri@gmail.com
University of Monastir
Chokri Ben Salah
University of Monastir

Research Article

Keywords: model-driven engineering, digital twin, metamodeling, software automation, runtime assurance, cyber-physical security

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11076427/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Automated cross-layer metamodeling for secure digital twins: attack detectability by construction

Ferdaws Ben Naceur[1], Mahmoud El Amri[2*] and Chokri Ben Salah[1,3]

[1] *Laboratory of Advanced Technology and Intelligent Systems, (LATIS), University of Sousse, Sousse 4023, Tunisia*

[2] *Laboratory of Automation, Electrical Systems and Environment (LASEE), University of Monastir, Monastir, Tunisia*

[3]*Higher Institute of Applied Sciences and Technology of Sousse (ISSAT Sousse), University of Sousse, Sousse, Tunisia*

* *Corresponding author. E-mail:*mahmoudelamri@gmail.com

ARTICLE INFO

Keywords:
model-driven engineering
digital twin
metamodeling
software automation
runtime assurance
cyber-physical security

ABSTRACT

This paper presents an automated model-driven engineering approach for constructing secure digital twins of cyber-physical systems. The core contribution is a cross-layer metamodel that represents physical assets, information flows, AI analytics, optimization services and cyber threats as typed, interrelated software artifacts within one conformance-checked structure. A conforming model is instantiated as an executable digital twin, so security rules can access not only measured values but also information-layer properties such as packet sequence, timestamp and integrity metadata. The approach is evaluated through a smart-grid digital twin with photovoltaic generation, battery storage, residential demand and electric-vehicle charging, using eleven years of ERA5-Land data (2015–2025). The executable services combine multi-horizon neural forecasting, linear-programming dispatch and cross-layer anomaly detection. Forecasting achieves 6.8–10.6% normalized RMSE and the dispatch reaches a 45% renewable fraction. Under staged false-data-injection, denial-of-service and replay attacks, the value-only detector attains 0.92 ROC-AUC but detects no replay attacks, whereas the metamodel-enabled information-integrity check raises replay detection to 100% and recovers about 81% of the attack-induced operating cost. The results show that explicit cross-layer software models can make classes of cyber manipulation detectable by construction that remain invisible to value-space monitoring alone.

Nomenclature and Acronyms

Acronym	Definition	Acronym	Definition
AI	Artificial Intelligence	LP	Linear Program
AAS	Asset Administration Shell	MDE	Model-Driven Engineering
AR(1)	First-order Autoregressive	MDT	Model-Driven Digital Twin
AUC	Area Under the ROC Curve	MLP	Multi-Layer Perceptron
CPS	Cyber-Physical System	MOF	Meta-Object Facility
DoS	Denial of Service	MPC	Model Predictive Control
DT	Digital Twin	NOCT	Nominal Operating Cell Temperature
EMS	Energy Management System	NWP	Numerical Weather Prediction
EV	Electric Vehicle	PV	Photovoltaic
FDI	False-Data Injection	ROC	Receiver Operating Characteristic
GHI	Global Horizontal Irradiance	RMSE	Root-Mean-Square Error
IoT	Internet of Things	SOC	State of Charge
ISO	Int. Org. for Standardization	ToU	Time-of-Use

Principal symbols

Symbol	Definition	Symbol	Definition
G	Plane-of-array irradiance [W/m²]	T_{cell}, T_{amb}	Cell / ambient temperature [°C]
P_{pv}, P_{load}	PV / load power [kW]	E_t, SOC_t	Battery energy [kWh] / state of charge
η_c, η_d	Charge / discharge efficiency	z_t / $\hat{z}_t$	Measurement / twin estimate
r_t	Residual	S_t	Detection statistic
a_t	Attack vector	π_t	Time-of-use tariff [$/kWh]
Δt	Time step [h]	h	Forecast horizon [h]

1. Introduction

Digital twins have evolved from descriptive virtual replicas into software-intensive runtime systems that ingest streams, maintain executable state, invoke analytics and optimization services, and issue decisions back to cyber-physical assets. Recent energy-oriented reviews emphasize the growing coupling of digital twins with machine learning, renewable integration, storage and vehicle-to-grid operation [1, 2]. From a software-engineering perspective, this evolution is more fundamental than a change of application domain: the twin becomes a long-lived software artifact whose correctness depends on the consistency of models, runtime data, interfaces and executable services. Conceptual work on digital twins therefore increasingly treats the twin as an organized set of heterogeneous models and synchronized data rather than as a single simulation [3, 4].

This software-intensive view exposes problems that are familiar to automated software engineering: how to represent heterogeneous artifacts in a machine-processable form, how to check their structural consistency, how to trace runtime decisions back to model elements, and how to evolve the system without losing semantic relationships. Multi-paradigm modelling has been proposed for cyber-physical systems precisely because electrical, communication, software and control concerns cannot be represented faithfully by one narrow formalism [5]. A recent systematic mapping of model-driven engineering (MDE) for digital twins confirms that metamodels, transformations and synchronization mechanisms are increasingly used to structure twin engineering, but it also reports considerable fragmentation in languages, tools and lifecycle coverage [6].

Model-driven approaches have begun to close the gap between engineering models and executable digital twins. SysML v2 and Asset Administration Shell (AAS) transformations make system structure reusable across engineering and runtime representations [7], while SysML-based approaches formalize the construction of twinned assets and their interfaces [8, 9]. The AAS literature further shows that a digital twin requires more than a data container: different AAS types support different levels of lifecycle integration, semantics and executable behavior [10]. In parallel, work at the intersection of MDE and AI shows that domain-specific languages and model transformations can reduce the complexity of AI software engineering, although support remains concentrated on particular lifecycle stages rather than on an end-to-end engineering process [11].

These advances are important, but they leave a recurring boundary problem. Most digital-twin metamodels focus on assets, geometry, behavior, data interfaces or lifecycle exchange. AI components are frequently attached as services after the structural model has been defined, and threat assumptions are commonly specified outside the same metamodel. Consequently, a model may describe what is measured and controlled without representing how a packet carries sequence and time semantics, which analytic service consumes it, or which attack can alter that information. This separation weakens traceability exactly where runtime assurance requires it most.

Recent work in Automated Software Engineering provides several mechanisms that are directly relevant to this problem. Lano and Siala use MDE to automate software-language translation, illustrating how explicit metamodels and transformation rules can turn structural knowledge into repeatable software automation [12]. Bernardi et al. integrate model-driven simulation into design-stage performance assessment [13]. Meyer et al. show that machine-readable specifications improve the portability and automation of scenario-based testing across execution platforms [14]. Mazkatli et al. address continuous consistency between architectural performance models and evolving software artifacts [15]. Nguyen et al. combine digital-twin context with automated security orchestration for critical infrastructures [16], and Ghasemi et al. use graph-based system models as the basis for automated model-based testing with deep reinforcement learning [17]. Together, these studies demonstrate the value of machine-readable models, executable semantics and automated consistency mechanisms. However, none of these mechanisms alone addresses the specific cross-layer question considered here: whether a single metamodel can make cyber manipulations detectable by ensuring that the required evidence exists in the runtime software structure.

AI forecasting and optimization have become standard services in smart-energy software. Recent work reports accurate hybrid deep-learning models for residential energy forecasting [18], demand-side valley filling for electric-vehicle charging [19], safety-aware deep reinforcement learning for charging coordination [20], and open simulation environments for benchmarking smart charging and vehicle-to-grid strategies [21]. These methods improve prediction or control performance, but their software boundaries are often implicit. Input provenance, packet integrity, feature contracts, failure semantics and the relationship between an AI output and the physical component it controls are typically encoded in implementation logic rather than in a shared system model.

This algorithm-centric organization creates two engineering risks. First, reproducibility depends on assumptions that are scattered across code, preprocessing scripts and experiment configuration. Second, security analysis sees the AI service and the physical process through a reduced measurement interface. A manipulated value can therefore remain numerically plausible while its information-level provenance is invalid. In such a design, improving the predictor or anomaly classifier does not recover evidence that the software architecture discarded before detection was performed.

Smart-grid cybersecurity has moved from classical residual tests toward distributed and learning-based detection. Federated deep learning has been used to identify false-data injection while reducing centralized data exposure [22]; temporal multi-graph convolutional networks exploit spatiotemporal dependencies [23]; hybrid deep models distinguish attacks from physical disturbances [24]; and recent reviews synthesize progress in attack detection and localization [25]. Physics-guided learning has also been applied to electric-vehicle cyberattack detection [26], while AI-based intrusion detection has been developed for charging stations [27]. At the communication layer, current reviews emphasize that packet loss, latency, interoperability and IoT reliability are integral to smart-grid operation rather than external networking details [28, 29].

Digital-twin cybersecurity research broadens this view but remains heterogeneous. Reviews of AI-enabled security twins identify the potential of synchronized virtual models for anomaly detection and predictive defense [30], and recent surveys classify the models, tools and security applications used by security-enhancing twins [31]. Blockchain-backed digital-twin frameworks seek to protect provenance and trust across cyber-physical interactions [32]. Lifecycle studies argue that twins can support security assessment from design through operation [33]. Formal approaches such as CRYSTAL derive runtime monitors from verified reference models and use a compact digital twin for attack detection [34]. A recent IEEE Access review similarly identifies persistent challenges in digital-twin attack surfaces, data protection and cross-layer security [35].

The critical limitation across these streams is not the absence of powerful detectors. It is the separation between the software model of the system and the evidence used by the detector. Security frameworks often place an orchestration or monitoring layer around a twin, whereas energy-management studies typically place forecasting and optimization above a measurement stream. In both cases, the semantics of the information layer may remain external to the executable model. Replay attacks illustrate the consequence: a replayed value can satisfy physical bounds and resemble a legitimate historical trajectory, so a value-space detector has no basis for distinguishing it unless freshness, sequence or provenance is represented explicitly.

The literature therefore reveals four connected gaps. First, digital-twin engineering remains structurally fragmented across physical, information, AI and security artifacts [3, 6, 10]. Second, MDE approaches improve machine-processable structure and automation, but AI services and cyber threats are not consistently represented as first-class elements of the same metamodel [7, 11, 12]. Third, advanced attack detectors improve classification accuracy but frequently operate on value streams after information-layer semantics have already been abstracted away [22-27]. Fourth, security-oriented twins provide valuable context and orchestration, yet the detectability of an attack is rarely traced to a conformance rule that guarantees the availability of the required runtime evidence [16, 30-35].

This paper addresses these gaps as an automated software-engineering problem. The objective is not to propose another forecasting architecture or another value-space anomaly classifier. Instead, we define a cross-layer metamodel whose conformance rules require energy components, data packets, AI services and threat actions to be represented through explicit typed relationships. The central claim is structural: if information-layer properties such as sequence and timestamp are first-class model attributes and the runtime twin conforms to that model, then security rules can use those properties by construction. The empirical case study tests this claim by comparing value-only detection with model-enabled cross-layer detection under false-data injection, denial-of-service and replay scenarios.

The paper makes four contributions. (1) It defines a cross-layer metamodel that unifies physical assets, information flows, AI analytics, optimization services and cyber threats within one machine-processable structure. (2) It instantiates executable semantics from the conforming model, preserving traceability from metamodel elements to sensing, forecasting, dispatch and security services. (3) It derives cross-layer runtime assurance rules from represented attributes rather than adding security evidence ad hoc after implementation. (4) It evaluates the resulting software artifact on a reproducible smart-grid case study using eleven years of ERA5-Land data and an explicit multi-attack campaign, while separating real reanalysis inputs from synthetic calibrated demand and mobility data.

The evaluation is organized around three research questions. RQ1 asks whether one metamodel can represent and connect the physical, information, AI and threat concepts needed by the executable twin. RQ2 asks whether explicit information-layer semantics change the detectability of attacks that remain plausible in value space. RQ3 asks whether model-enabled detection produces a measurable benefit when propagated through the closed-loop dispatch service. These questions place model expressiveness, automation and runtime assurance before domain-specific energy performance, which is used as the evaluation environment rather than as the primary contribution.

The remainder of the paper is organized as follows. Section 2 presents the automated MDE workflow. Section 3 defines the cross-layer metamodel and conformance rules. Sections 4 to 7 describe executable component semantics, forecasting, dispatch and security rules. Section 8 presents the empirical protocol and data. Section 9 reports the evaluation results. Section 10 discusses validity and provenance, and Section 11 concludes the paper.

2. Automated Model-Driven Engineering Workflow

Figure 1 shows the MDT framework as an automated workflow over eight stages. A physical smart grid is observed through typed IoT/data-acquisition artifacts. Their streams synchronize a digital twin whose structure conforms to the system metamodel. Forecasting, cybersecurity and optimization services operate on that synchronized state and produce control decisions. The automation boundary is explicit: the current prototype performs structural conformance checking, model instantiation, state synchronization, forecasting, linear-programming dispatch and rule-based cross-layer detection; it does not claim automatic synthesis of all implementation code from the metamodel.

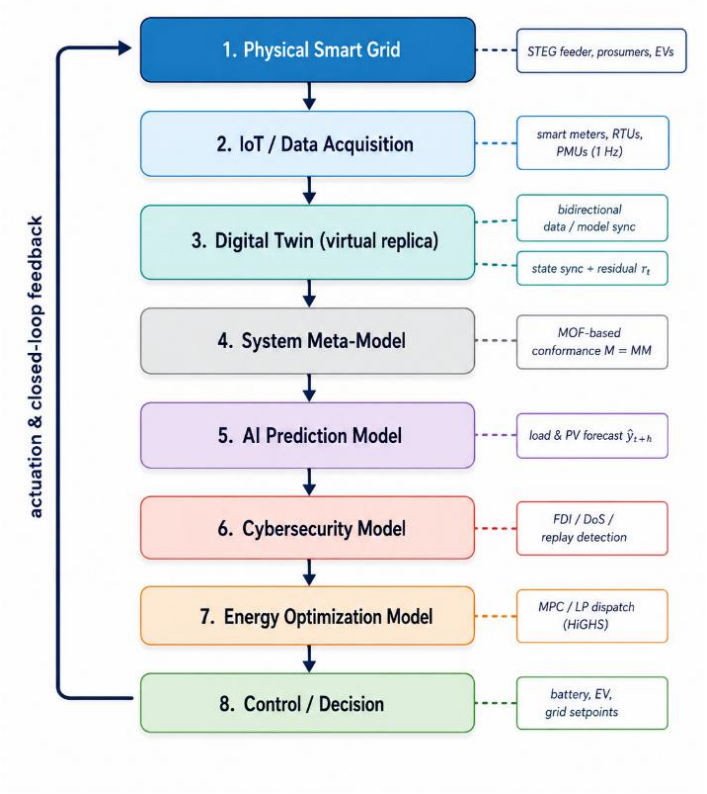


Fig. 1. Model-Driven Digital-Twin framework and closed-loop workflow.

The architecture makes layer boundaries and data contracts explicit: each stage consumes and produces artifacts whose structure is defined once in the metamodel. This permits the security service to reason about information-layer properties, such as sequence numbers and timestamps, that would otherwise be discarded by a value-only pipeline. Following the Meta-Object-Facility discipline, a running configuration is accepted only when it conforms to the metamodel:

$$M \vDash MM \tag{1}$$

a relation we check structurally before the twin is instantiated.

3. Cross-Layer Metamodel and Conformance Rules

The metamodel in Figure 2 defines the software structure used in the study. It organizes the domain into four layers and makes cross-layer dependencies explicit. The resulting typed graph defines the abstract syntax expected by the executable twin and the information available to runtime services.

Energy/physical layer. Classes SmartGrid, PV Array, Battery, EV/Station and SmartHome carry the physical attributes and states of Section 4 (e.g. Battery.SOC, PV.P_ac). They are the twinned assets.

Information layer. Classes IoTSensor, CommLink and DataPacket model the acquisition and transport of measurements. A DataPacket carries not only a value but a timestamp, a sequence number seq and an integrity field. By treating the packet's metadata as first-class attributes, the metamodel makes information-layer integrity a checkable property rather than an implicit assumption. IoTSensor measures an energy-layer component; CommLink transmits; DataPacket encapsulates the reading.

AI/analytics layer: Classes Forecaster, StateEstimator, AnomalyDetector and Optimizer model the decision logic. The Forecaster feeds predictions to the Optimizer; the StateEstimator produces the residual r_t the AnomalyDetector consumes; the Optimizer controls the energy layer.

Threat layer: Classes Attacker, FDIAttack, DoSAttack and ReplayAttack model the adversary. Each threat class targets a specific information- or analytics-layer element: FDIAttack injects bias into a DataPacket.value; DoSAttack drops packets on a CommLink; ReplayAttack overwrites DataPacket.seq/timestamp with a stale window. Placing the attacker in the model, with typed associations to the elements it can reach, makes potential failure modes an explicit and enumerable part of the system description.

The inter-layer relations connect the four layers and act as traceability links. A ReplayAttack instance targets a DataPacket whose sequence/timestamp metadata becomes stale while its measured value can remain physically plausible. A detector restricted to value cannot observe this condition; a detector bound to the modeled metadata can. The metamodel therefore constrains not only system structure but also the evidence available to security automation. Section 7 operationalizes this relation by allowing the AnomalyDetector to consume energy-, information- and physics-layer attributes. Conformance ensures that a deployed configuration exposes the required attributes before the executable twin is instantiated.

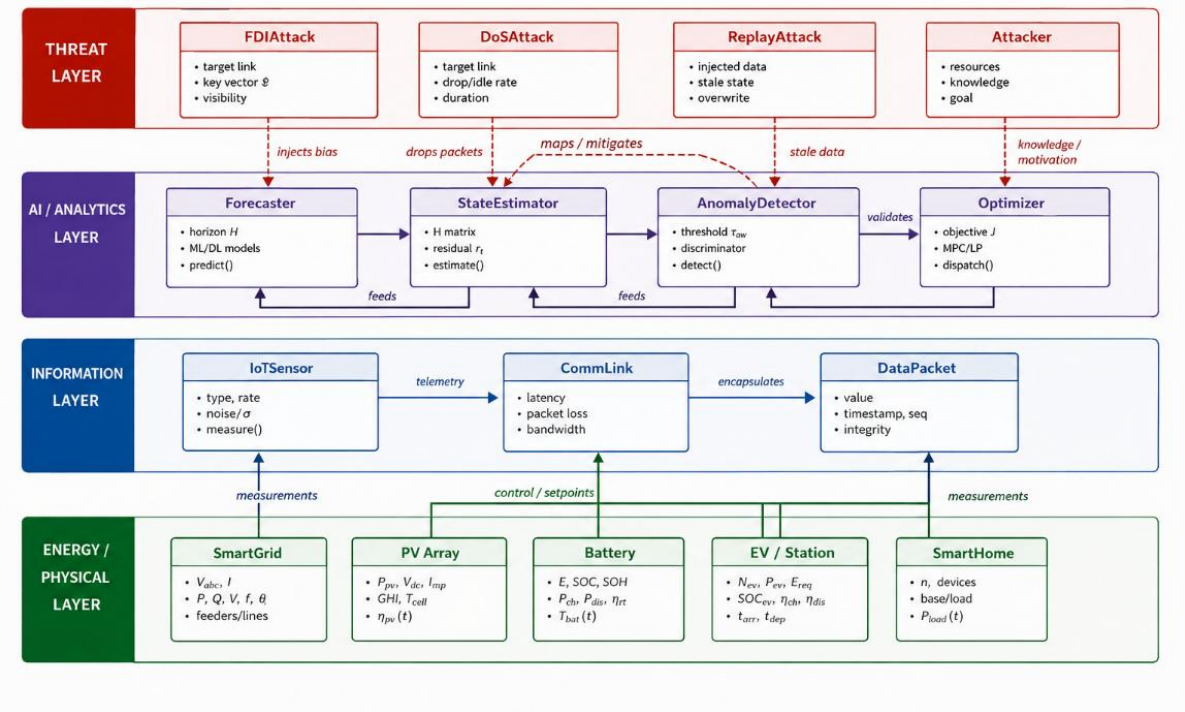


Fig. 2. Cross-layer metamodel of the cyber-physical energy system.

4. Executable Semantics of the Domain Model

The metamodel becomes operational through executable semantics attached to its domain concepts. The following models define the behavior used by the prototype; parameters are reported in Table 1 (Section 8). Time is discretized at the simulation step. These equations are case-study semantics, whereas the metamodel and typed relations provide the reusable software structure.

4.1 Photovoltaic Generation

The PV array follows the operational single-diode-derived power model of Villalva et al. [36] with a NOCT thermal correction. The cell temperature is

$$T_{cell} = T_{amb} + \frac{NOCT-20}{800}\, G \tag{2}$$

Where G is the plane-of-array irradiance [$\mathrm{W/m^2}$] obtained from the ERA5-Land surface solar radiation (Section 8) and $NOCT$ is the nominal operating cell temperature. The AC power delivered by an array of rated power P_{stc} is

$$P_{pv} = \min\left(P_{stc}\, \frac{G}{G_{stc}}\, [\,1 + \gamma\,(T_{cell} - 25)\,]\, \eta_{sys},\ P_{inv}\right) \tag{3}$$

With reference irradiance $G_{stc} = 1000\ \mathrm{W/m^2}$, temperature coefficient $\gamma < 0$, balance-of-system efficiency η_{sys}, and inverter clipping at rated P_{inv}.

4.2 Battery Storage

The battery uses the dynamic energy-reservoir model of Tremblay and Dessaint [37], with state-of-charge limits following Hannan et al. [38]. Charging and discharging power are represented explicitly in the state equation below. The stored energy evolves as:

$$E_{t+1} = E_t + \left(\eta_c\, P_{c,t} - \frac{P_{d,t}}{\eta_d}\right) \Delta t \tag{4}$$

And the state of charge is $SOC_t = E_t / E_{cap}$, constrained by

$$SOC_{\min} \le SOC_t \le SOC_{\max}, \qquad 0 \le P_{c,t},\, P_{d,t} \le P_{bat}^{\max} \tag{5}$$

Cycle ageing is charged linearly on energy throughput,

$$C_{deg,t} = c_{deg}\left(P_{c,t} + P_{d,t}\right) \Delta t \tag{6}$$

with c_{deg} a per-kWh wear cost.

4.3 Smart-Home Load

The aggregate demand of the residential cluster is represented by a calibrated stochastic profile consistent with the deep-learning load-modelling literature [18, 39]. The aggregate load of the modeled home cluster is:

$$P_{load,t} = n_h\, \bar{p}\, [\,\beta_d(t)\, \sigma_s(t)\,] + \varepsilon_t \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma_\ell^2) \tag{7}$$

Where $\bar{p}$ is the mean per-home power, $\beta_d(t)$ a two-peak diurnal shape (morning and evening), $\sigma_s(t)$ a seasonal factor, and ε_t aggregation noise. The profile is synthetic but range-calibrated (Section 10).

4.4 Electric-Vehicle Charging Station

EV sessions follow the mobility-model approach of Wan et al. [40] and the valley-filling formulation of Souza et al. [19]. For each charging session, the arrival time, departure time, initial state of charge and required energy are represented by:

$$E_i^{req} = \left(SOC_i^{dep} - SOC_i^{a}\right) E_{ev} \tag{8}$$

Delivered by chargers of power P_{chg} subject to the station limit

$$\textstyle\sum_{i \in \mathcal{A}(t)} P_{ev,i,t} \le P_{sta}^{\max} \qquad \sum_{t=t_i^a}^{t_i^d} P_{ev,i,t}\ \Delta t = E_i^{req} \tag{9}$$

Where $\mathcal{A}(t)$ is the set of connected vehicles at t. The equality constraint guarantees each vehicle is fully served by departure while leaving the *timing* free for the optimizer to fill demand valleys.

4.5 IoT Sensing and Communication Channel

Measurements reach the twin through the IoT channel of the information layer, modelled after the smart-grid communication reviews of Qays et al. [28] and Khan et al. [29]. A sensor observing physical quantity reports

$$z_t = x_t + \nu_t, \qquad \nu_t \sim \mathcal{N}(0, \sigma_m^2) \tag{10}$$

And each DataPacket is delivered only with probability $1 - \rho$, where ρ is the packet-loss rate; a lost packet leaves a gap in the sequence seq. In matrix form over the sensor set, $\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \boldsymbol{\nu}_t$, the standard linear measurement model on which the attacks of Section 7 operate.

4.6 Time-of-Use Tariff

Grid exchange is priced by a Tunisian-STEG-style time-of-use schedule with off-peak, mid and peak bands,

$$\pi_t = \begin{cases} \pi_{off} & t \in \mathcal{T}_{off} \\ \pi_{mid} & t \in \mathcal{T}_{mid} \\ \pi_{peak} & t \in \mathcal{T}_{peak} \end{cases} \tag{11}$$

With a flat export price π_{exp} for surplus PV.

4.7 Digital-Twin State and Residual

The digital twin maintains a nominal estimate of each measured quantity from the synchronized state and the AI forecaster (Section 5). Following the DT-with-ML paradigm [5] the twin residual is

$$r_t = z_t - \hat{z}_t \tag{12}$$

And its variance σ_i^2 per channel is estimated from attack-free operation as the empirical forecast-error variance. The residual is the raw signal for the energy-layer detector of Section 7; $\hat{z}_t$ is the *forecaster output*, not an oracle, so the twin carries realistic model error (Section 10).

5. AI Forecasting Service

Load and PV are forecast by multi-horizon multi-layer perceptrons (MLPs), the feed-forward counterpart to the recurrent load models of Kong et al. [39] and the hybrid smart-home predictors of Ou Ali et al. [18]. For each prediction target, the feature vector at each time step stacks the preceding lags and calendar encodings:

$$\mathbf{x}_t = [y_{t-1}, \dots, y_{t-L}, \sin(2\pi h_t/24), \cos(2\pi h_t/24), \sin(2\pi d_t/7), \cos(2\pi d_t/7)]^\top \tag{13}$$

Augmented, for PV, by an exogenous numerical-weather-prediction (NWP) irradiance forecast $\hat{G}_{t+h}$ rather than the true future irradiance, so that no look-ahead leakage inflates accuracy (Section 10). A separate network per horizon $h \in \{1, \dots, H\}$ maps

$$\hat{y}_{t+h} = f_{\theta_h}(\mathbf{x}_t) = W_2^{(h)}\, \phi\big(W_1^{(h)}\mathbf{x}_t + b_1^{(h)}\big) + b_2^{(h)} \tag{14}$$

With ϕ the ReLU activation, trained by minimizing the mean-squared error

$$\mathcal{L}(\theta_h) = \tfrac{1}{N}\textstyle\sum_t (y_{t+h} - \hat{y}_{t+h})^2 \tag{15}$$

Accuracy is reported as the normalized RMSE and the skill score against a persistence baseline $\hat{y}_{t+h}^{per} = y_t$,

$$nRMSE_h = \tfrac{1}{\bar{y}}\sqrt{\tfrac{1}{N}\textstyle\sum_t (y_{t+h} - \hat{y}_{t+h})^2} \qquad Skill_h = 1 - \tfrac{RMSE_h}{RMSE_h^{per}} \tag{16}$$

Models are trained on 2015–2023 and tested on 2024–2025 with a strict temporal split.

6. Automated Dispatch Service

The energy-management system computes a cost-optimal daily dispatch by linear programming, using the tractable convex core of the predictive micro-grid controller of Parisio et al. [41]. Over each optimization horizon, the decision vector contains grid import and export, battery charge and discharge, aggregate EV charging power, and a global peak-import epigraph variable. The objective minimizes energy cost net of export revenue, battery wear and a peak penalty that induces valley filling:

$$\min \sum_{t=1}^{T} \left[\pi_t P_t^{imp} \Delta t - \pi_{exp} P_t^{exp} \Delta t + c_{deg} \left(P_{c,t} + P_{d,t} \right) \Delta t \right] + \lambda_{pk} P^{pk} \tag{17}$$

The dispatch respects the nodal power balance, the battery dynamics of Section 4.2, the EV energy requirement of Section 4.4, the peak definition and the feeder limits:

$$P_t^{imp} + P_{pv,t} + P_{d,t} = P_{load,t} + P_{c,t} + P_t^{ev} + P_t^{exp} \tag{18}$$

$$P^{pk} \geq P_t^{imp}, \quad 0 \leq P_t^{imp} \leq P_{\max}^{imp}, \quad 0 \leq P_t^{exp} \leq P_{\max}^{exp} \tag{19}$$

The program is solved day-by-day with the battery state carried across days, using the HiGHS solver. Because the EV constraint fixes only the *total* energy per session and not its timing, the peak term $\lambda_{pk} P^{pk}$ shifts charging into low-demand, low-price hours, producing the valley-filling behaviour shown in Section 9.

7. Security Rules and Cross-Layer Detection

7.1 Threat Model as Typed Model Elements

Consistent with the CPS-security literature [42–44] the adversary can tamper with measurements traversing the information layer but cannot alter the physical plant. Three attack classes are modelled, each an instance of the corresponding threat-layer class of Section 3.

False-data injection (FDI) [42] adds a bias to the reported value,

$$z_t^a = z_t + a_t \tag{20}$$

Two FDI profiles are evaluated: a slow stealthy ramp, in which the injected attack vector increases gradually to evade abrupt-change tests, and an intermittent step attack. Denial of service (DoS) [28] drops all packets on a communication link for a sustained interval, leaving a sequence gap at the twin. Replay [44] overwrites the current window with a previously recorded, attack-free window while re-stamping the packet. The replayed values remain *physically plausible, but their sequence and timestamp metadata are stale.*

7.2 Model-Derived Cross-Layer Detection Rules

The detector is implemented as a runtime service whose inputs are determined by the metamodel. It fuses evidence from three represented layers. At the energy layer it forms the normalized-residual statistic, following classical bad-data detection principles [42, 43]:

$$S_t = \sum_i \left(\frac{r_{i,t}}{\sigma_i} \right)^2, \qquad \text{flag}_t^E = \mathbb{1}\{S_t > \tau\} \tag{21}$$

The detection threshold is calibrated to a 5 % false-positive rate on clean data. At the information layer, packet integrity is checked directly using the metadata reified by the metamodel:

$$\text{flag}_t^I = \mathbf{1}\{\text{seq}_t \text{ missing}\} \vee \mathbf{1}\{\tau_t^{\text{stamp}} \leq \tau_{t-1}^{\text{stamp}} \vee \text{nonce}_t \in H\} \tag{22}$$

Where $\mathcal{H}$ is the set of already-seen nonces. At the **physics layer** it rejects values outside operational bounds,

$$\text{flag}_t^P = \mathbb{1}\{P_{pv,t} \notin [0, P_{pv}^{\max}]\} \tag{23}$$

The combined decision is the union

$$\text{flag}_t = \text{flag}_t^E \vee \text{flag}_t^I \vee \text{flag}_t^P \tag{24}$$

Flagged samples are replaced by the twin nominal before dispatch (secure mode), whereas the naive mode trusts incoming measurements. The information-integrity rule is part of the model rather than an additional hidden assumption: its required fields belong to the DataPacket type. Consequently, replay can be checked whenever the model conforms, while a value-only architecture cannot recover information it does not represent.

8. Empirical Evaluation

8.1 Evaluation Protocol, Site and Data

The framework is evaluated as an executable software artifact. RQ1 is assessed by instantiating all four metamodel layers and checking that the runtime configuration exposes the typed cross-layer relations required by the services. RQ2 is assessed by comparing value-only and cross-layer detection under FDI, DoS and replay. RQ3 is assessed by propagating detected and undetected manipulations through dispatch and measuring cost and constraint effects. The validation context is a coastal micro-grid in Sousse, Tunisia. Solar and thermal drivers come from ERA5-Land reanalysis over 2015–2025 (96,432 hourly steps). The de-accumulation convention for surface solar radiation was checked programmatically before conversion to irradiance.

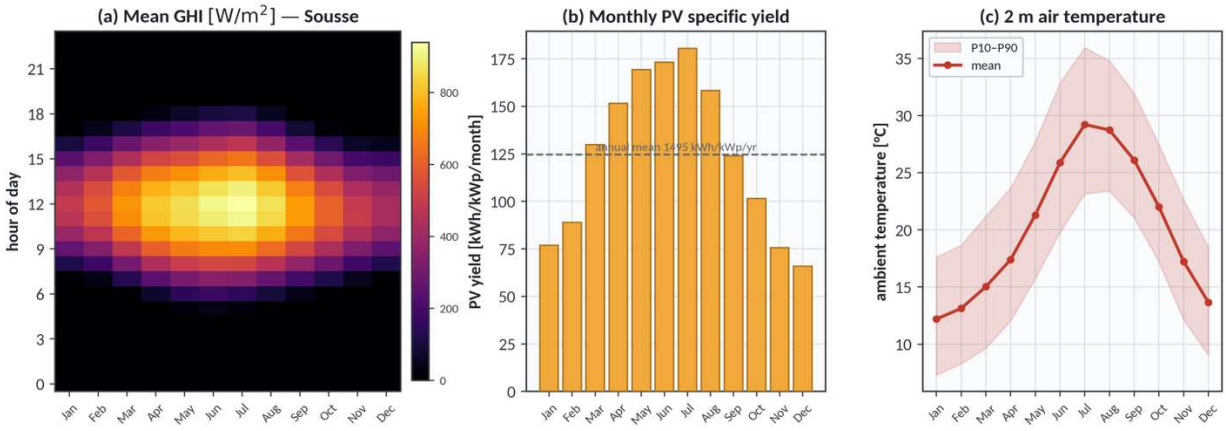


Fig. 3. Sousse solar and thermal resource characterization (2015-2025).

The residential-load and EV-mobility series are synthetic but range-calibrated to realistic operating ranges; they are used to exercise the software workflow rather than to certify a specific installation. The tariff follows a representative STEG-like time-of-use structure. Figure 4 shows one summer day and the interaction among PV, residential demand, deferrable EV charging and tariff periods. Section 10 separates real, synthetic and representative inputs to make the evaluation boundary reproducible.

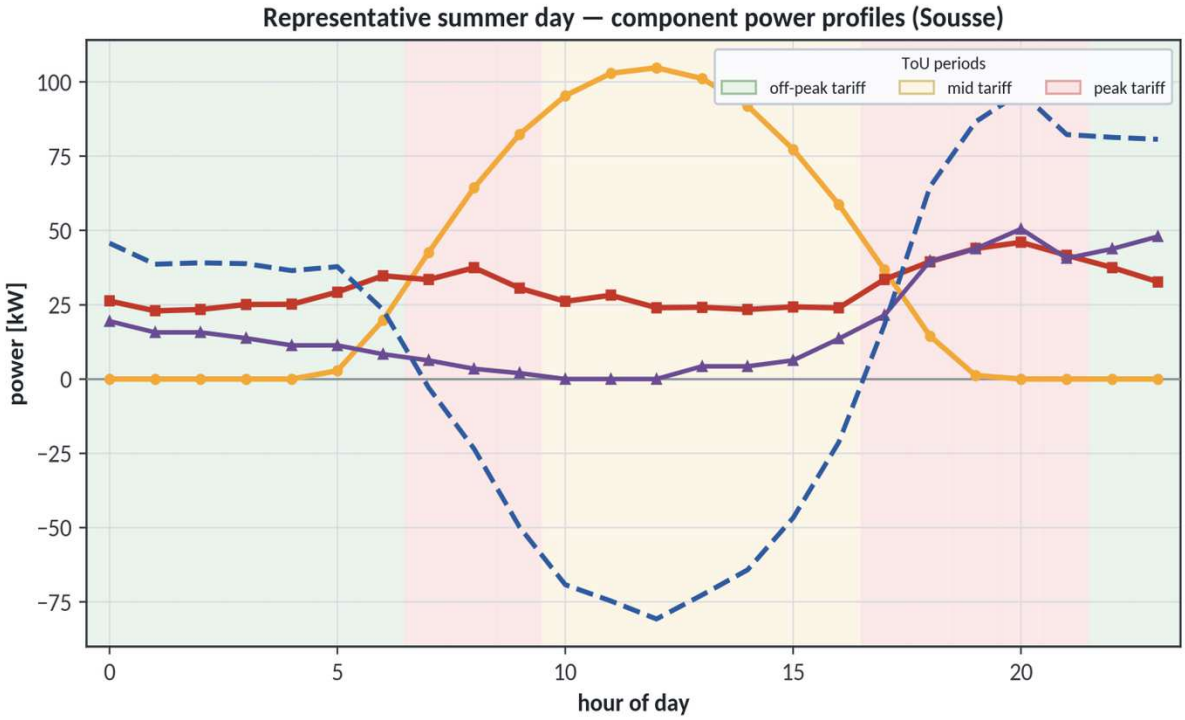


Fig. 4. Representative summer-day power profiles and ToU periods.

8.2 System Parameters

The numerical configuration used to instantiate the executable case study is summarized in Table 1. It brings together the rated capacities, operating limits, efficiencies, and model settings of the PV array, battery, smart-home aggregate, EV charging station, feeder, and forecasting module. Making this parameterization explicit improves reproducibility and configuration traceability, and provides a consistent basis for the forecasting, dispatch, and cybersecurity experiments reported in the following sections.

Table 1. System parameters.

Component	Parameter	Value
PV array	P_{stc} / γ / η_{sys} / P_{inv}	150 kWp / -0.41 %/°C / 0.86 / 140 kW
Battery	E_{cap} / $P_{bat}^{\max}$ / $\eta_{c,d}$ / SOC / c_{deg}	200 kWh / 60 kW / 0.95 / 0.10–0.95 / 0.03 $/kWh
Smart home	homes / mean power	30 / ≈ 1 kW per home
EV station	vehicles/day / battery / charger / station cap	≈ 16 / 52 kWh / 11 kW / 80 kW
Feeder	$P_{\max}^{imp}$ / $P_{\max}^{exp}$	90 kW / 60 kW
Forecaster	lags L / horizon H / hidden	24 / 24 / (160, 96)

9. Evaluation Results and Discussion

9.1 Forecasting Accuracy

The neural forecaster (Figure 5) tracks both series closely over the test period. Load is predicted with a normalized RMSE rising only mildly from 6.8 % at one hour to 7.3 % at 24 hours, with an 84 % skill improvement over persistence

at six hours. PV prediction, driven by a noisy NWP irradiance forecast rather than the true future irradiance, shows a stronger horizon-dependent degradation (from 8.3 % to 10.6 % nRMSE) while still achieving 95 % six-hour skill; the annual specific yield is 1494 kWh/kWp. The mild horizon growth for load and the steeper, weather-limited growth for PV are consistent with the expected behaviour after removing look-ahead leakage (Section 10) and determine the residual variances σ_i used by the detector.

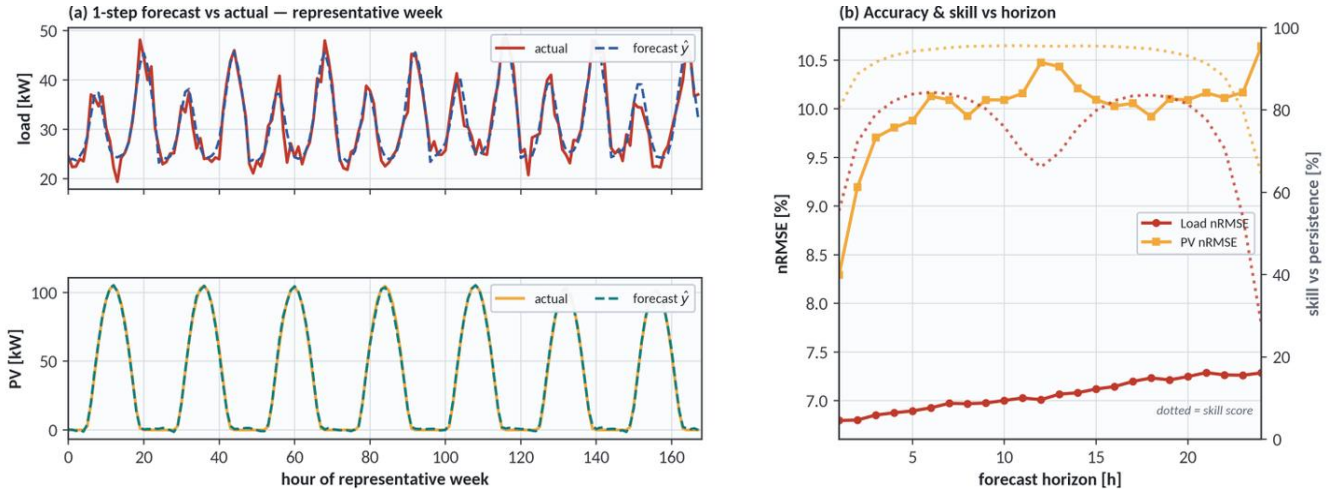


Fig. 5. Load and PV forecasting performance.

9.2 Economic Dispatch

Over the 731-day test horizon the LP dispatch (Figure 6) achieves a total operating cost of **$38,160** (≈ $52.2/day), of which $6,495 is battery wear. It draws **448.8 MWh** from the grid, exports 70.0 MWh of PV surplus and self-consumes **84.4 %** of the 449.5 MWh generated, yielding a **45.1 % renewable fraction** of the 817.2 MWh demand. The valley-filling penalty holds the peak import to **59.2 kW**, comfortably inside the 90 kW feeder limit; Figure 6 shows EV charging (negative battery/charge bars) migrating into the low-price overnight valley while the battery buffers the midday PV peak and the evening demand ramp. The SOC trajectory stays within its 10–95 % band throughout.

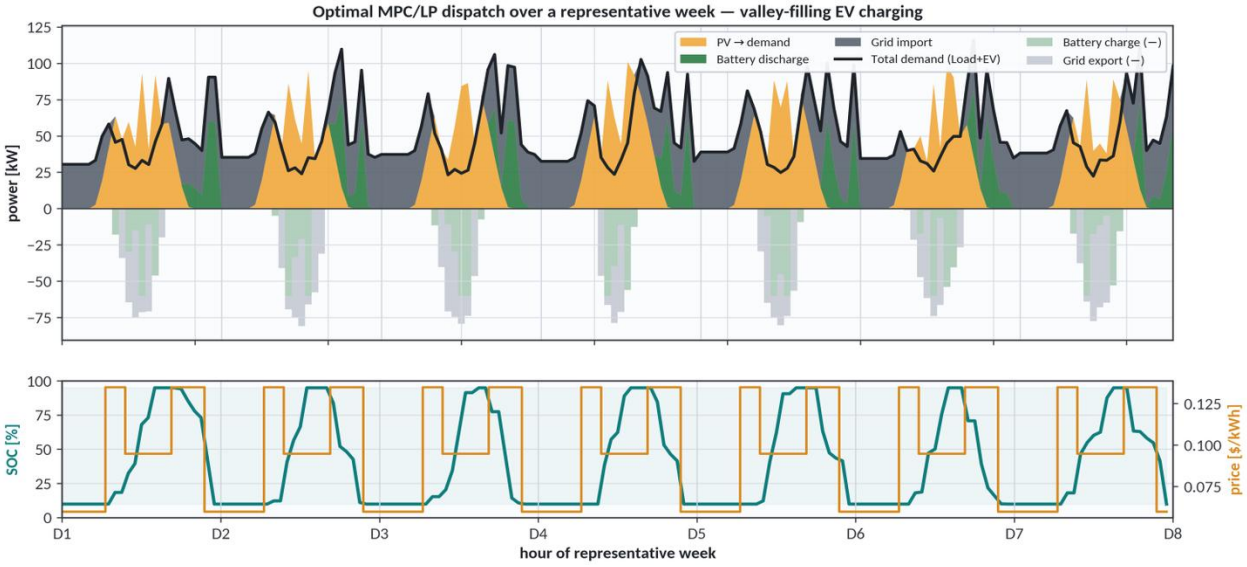


Fig. 6. Representative-week LP/MPC dispatch and battery SOC.

9.3 RQ2: Cross-Layer Attack Detectability

Figure 7 shows the 30-day attack campaign and Figure 8 the detection performance.

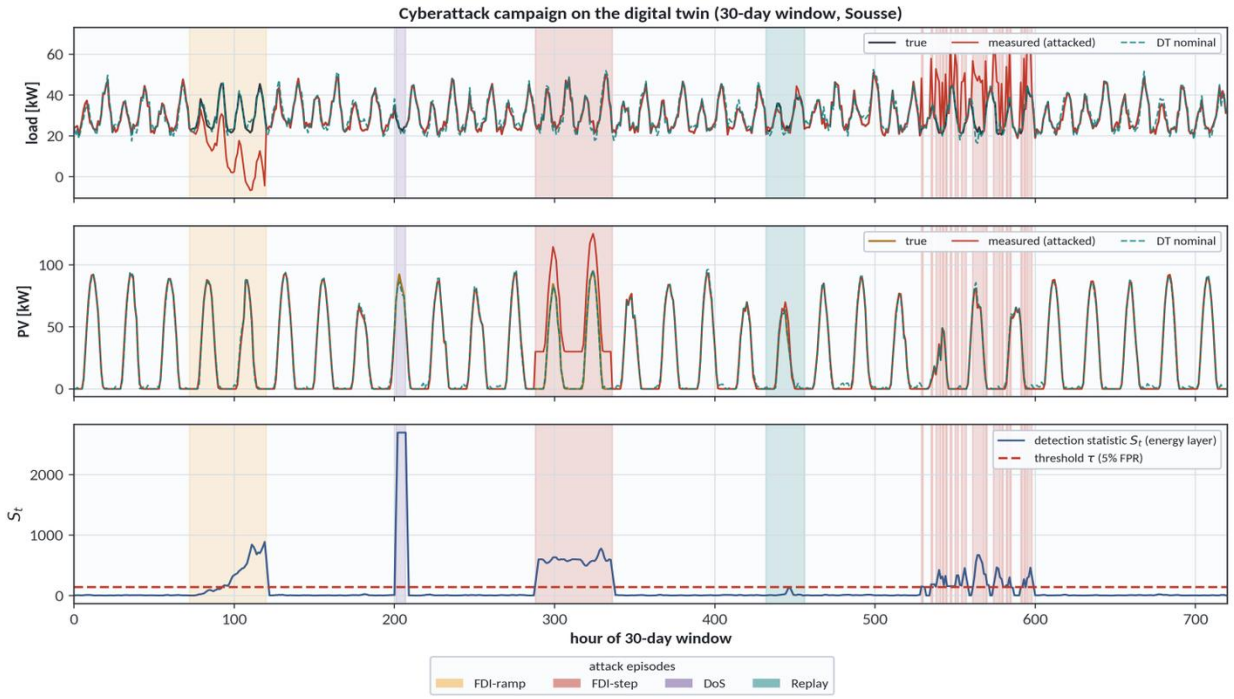


Fig. 7. Thirty-day cyberattack campaign and detection statistic.

The energy-layer residual detector attains a ROC-AUC of 0.922, and at a 5 % false-positive threshold the combined detector reaches an 85.5 % overall true-positive rate at 1.1 h mean latency. The per-attack breakdown (Figure 8b) quantifies the contribution of the cross-layer metamodel:

- **DoS** is detected with 100 % rate because the sustained sequence gap produces both a large residual and an information-layer flag.
- **Step-FDI** is detected at 97.5 %, its abrupt bias driving the residual well past threshold.
- **Ramp-FDI** is the most challenging case at **56.2 %**: the slow stealthy drift is designed to stay near the noise floor, and a value residual can only catch its later, larger portion. This partial rate reflects the intended stealthiness of the ramp attack under the stated detector settings.
- Replay attacks reveal the limitation of value-only detection. Because the replayed window is a physically plausible past day, it is invisible to the value residual, with 0 % detection (visible in Figure 7, where the statistic never crosses during the replay episode). The information-layer integrity check exposed by the metamodel, based on stale timestamps and repeated nonces, raises replay detection to 100 %.

The 0% to 100% replay-detection change answers RQ2: the gain is caused by the availability of modeled information-layer evidence rather than by a more complex value-space classifier. Because replay is constructed to preserve plausible values, value-space processing alone has no freshness signal to exploit. The cross-layer metamodel makes that signal part of the software contract.

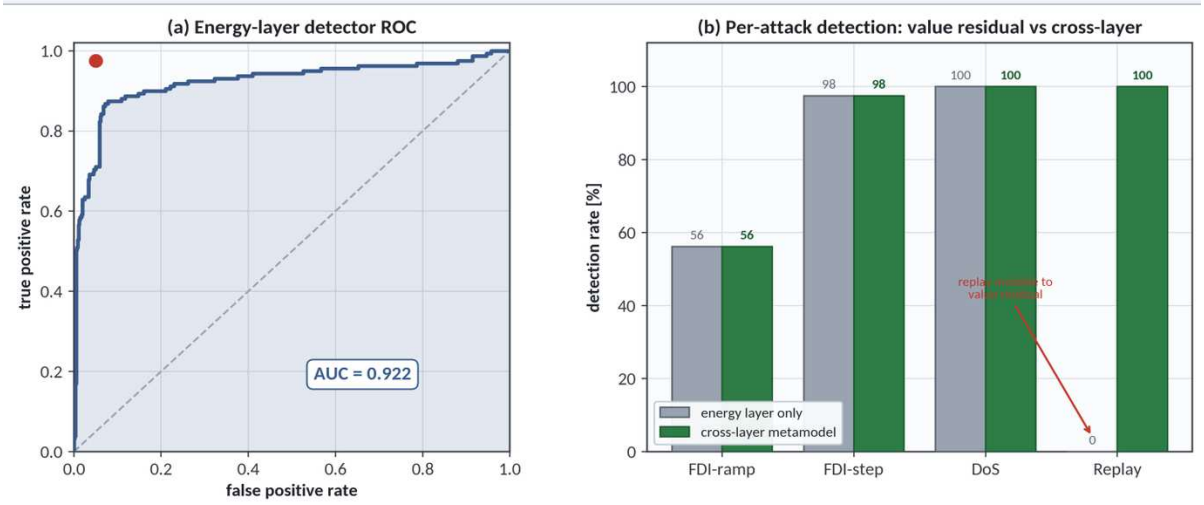


Fig. 8. Cross-layer attack-detection performance.

9.4 RQ3: Closed-Loop Operational Impact

Figure 9 propagates the attack campaign into the executable dispatch service. The naive configuration that trusts manipulated data incurs a +5.3% ($83) cost increase over the 30-day window and raises peak import from 45.5 to 52.6 kW. In secure mode, replacing flagged inputs by the twin nominal limits the increase to +1.0%, recovering 80.6% of attack-induced cost with no SOC or feeder violations. The remaining effect is associated mainly with stealthy ramp-FDI samples that remain below the detector threshold. These results answer RQ3 by showing the downstream consequence of exposing cross-layer evidence to runtime services.

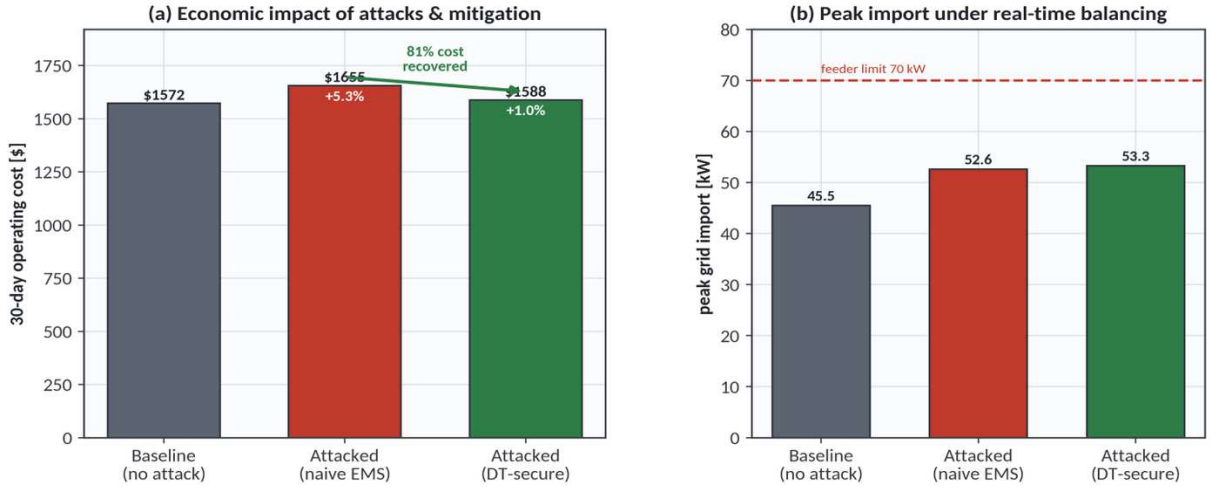


Fig. 9. Economic impact of attacks and DT-secure mitigation.

9.5 Robustness Across Weather Years

Repeating the annual dispatch across all eleven ERA5-Land years (Figure 10) shows the design is not tuned to a single favourable year. The renewable fraction is **45.2 ± 0.33 %** and the PV specific yield varies with a coefficient of variation of only **0.9 %** (mean 1495 kWh/kWp), reflecting the stability of the Tunisian solar resource; the seasonal PV availability ranges from 16.0 kW (DJF) to 34.8 kW (JJA). The economic and renewable-fraction conclusions therefore hold across a decade of real weather variability.

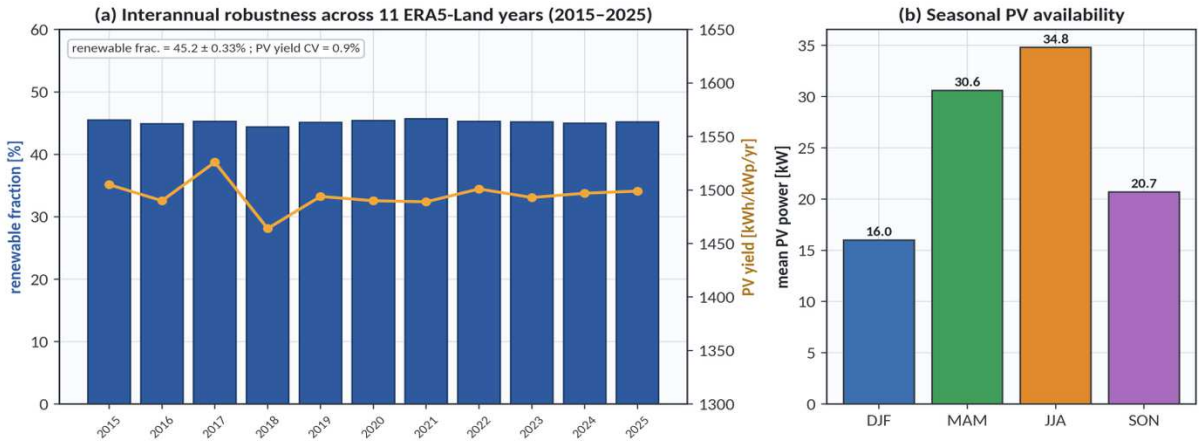


Fig. 10. Interannual renewable-performance robustness (2015-2025).

10. Threats to Validity and Data Provenance

Construct and internal validity. The main construct under evaluation is whether explicit cross-layer modeling changes the evidence available to automated detection. The replay experiment isolates this mechanism because replayed values remain plausible while packet freshness is violated. The detector threshold is calibrated on clean data, and the digital-twin nominal is the forecaster output plus empirical forecast-error noise rather than an oracle. External validity. ERA5-Land irradiance and temperature are reanalysis data for Sousse, whereas residential demand and EV mobility are synthetic but range-calibrated. The tariff is representative of a STEG-like time-of-use structure rather than a live tariff. Accordingly, absolute cost values are illustrative and should not be interpreted as site certification. The present evaluation uses one application domain and one executable prototype; generalization to other cyber-physical domains requires further replication. Reproducibility and limitations. The PV, battery, forecasting and optimization models are executable and the temporal train/test split is stated explicitly. The 0% value-only replay detection demonstrates a structural limitation of value-space monitoring under the defined threat model, while the 56% ramp-FDI rate remains a limitation of the current detector. Future work should provide a public artifact package, add cumulative residual tests for stealthy ramps, encode the metamodel in a standards-based representation such as SysML v2 or Asset Administration Shell, and validate the workflow on field measurements.

11. Conclusion

This paper presented an automated model-driven engineering approach in which a secure digital twin is instantiated from a cross-layer metamodel that explicitly relates physical assets, information packets, AI services, optimization and threats. The results show that the evidence available to a runtime security service depends on the information represented in the software model. In the executable case study, value-only monitoring achieved 0.922 ROC-AUC but detected no replay attacks; the information-integrity fields required by the metamodel made replay deterministically observable and increased replay detection from 0% to 100%. The resulting secure mode recovered about 81% of attack-induced cost while maintaining operational constraints. Forecasting and dispatch results confirm that the same model can support normal runtime services as well as security checks. The evaluation also identifies clear limits: synthetic demand and mobility inputs, one application domain, partial detection of stealthy ramp FDI, and no field deployment. Future work will formalize the metamodel in SysML v2/AAS, package conformance and transformation rules as reusable tooling, add cumulative detection for slow attacks, and evaluate portability across additional cyber-physical software systems.

Acknowledgements

The authors acknowledge the public data providers cited in the manuscript.

Statements and Declarations

Data Availability

ERA5-Land input data are publicly available through the Copernicus Climate Data Store. Processed data and analysis code supporting the reported results are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing financial or non-financial interests that are directly or indirectly related to this work.

Funding

The authors received no financial support for the research, authorship, or publication of this article.

Author Contributions

Conceptualization and methodology: M. El Amri, F. Ben Naceur and C. Ben Salah; formal analysis and data curation: M. El Amri and F. Ben Naceur; software and validation: M. El Amri and F. Ben Naceur; writing, original draft: M. El Amri; writing, review and editing: M. El Amri and C. Ben Salah. All authors reviewed and approved the final manuscript.

References

[1] Das, O., Zafar, S.B.Q., Sanfilippo, F., Rudra, S., Kolhe, M.L.: Advancements in digital twin technology and machine learning for energy systems: a review. Energy Conversion and Management: X 24, 100715 (2024).

[2] Al-Shetwi, A.Q., Atawi, I.E., El-Hameed, M.A., Abuelrub, A.: Digital Twin Technology for Renewable Energy, Smart Grids, Energy Storage and Vehicle-to-Grid Integration: Advancements, Applications, Key Players, Challenges and Future Perspectives in Modernising Sustainable Grids. IET Smart Grid 8(1), e70026 (2025).

[3] Eramo, R., Bordeleau, F., Combemale, B., van den Brand, M., Wimmer, M., Wortmann, A.: Conceptualizing digital twins. IEEE Software 39(2), 39–46 (2022).

[4] David, I., Archambault, P., Wolak, Q., Vu, C.V., Lalonde, T., Riaz, K., Syriani, E., Sahraoui, H.: Digital twins for cyber-biophysical systems: challenges and lessons learned. In: 26th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS), pp. 1–12 (2023).

[5] Amrani, M., Blouin, D., Heinrich, R., Rensink, A., Vangheluwe, H., Wortmann, A.: Multi-paradigm modelling for cyber-physical systems: a descriptive framework. Software and Systems Modeling 20(3), 611–639 (2021).

[6] Lehner, D., Zhang, J., Pfeiffer, J., Sint, S., Splettstößer, A.-K., Wimmer, M., Wortmann, A.: Model-driven engineering for digital twins: a systematic mapping study. Software and Systems Modeling 24, 1339–1377 (2025).

[7] Ferko, E., Berardinelli, A., Bucaioni, A., Behnam, M., Wimmer, M.: From engineering models to digital twins: generating AAS from SysML v2 models. Journal of Systems and Software 233, 112688 (2026).

[8] Ouari, N., Ioualalen, M., Hammad, A.: A SysML-based model-driven approach for digital twin engineering. In: Computer Science and its Applications, Lecture Notes in Networks and Systems 1145, Springer (2024).

[9] Wilking, M., Sauer, C., Schleich, B., Wartzack, S.: SysML 4 digital twins: utilization of system models for the design of digital twins. Proceedings of the Design Society 2, 1815–1824 (2022).

[10] Zhang, J., Ellwein, C., Heithoff, M., Michael, J., Wortmann, A.: Digital twin and the asset administration shell. Software and Systems Modeling 24, 771–793 (2025).

[11] Rädler, S., Berardinelli, L., Winter, K., Rahimi, A., Rinderle-Ma, S.: Bridging MDE and AI: a systematic review of domain-specific languages and model-driven practices in AI software systems engineering. Software and Systems Modeling 24, 445–469 (2025).

[12] Lano, K., Siala, H.: Using model-driven engineering to automate software language translation. Automated Software Engineering 31, 20 (2024).

[13] Bernardi, S., Gómez, A., Merseguer, J., Perez-Palacin, D., Requeno, J.I.: DICE simulation: a tool for software performance assessment at the design stage. Automated Software Engineering 29, 36 (2022).

[14] Meyer, M.-A., Zouari, M., Bannenberg, S., Deppe, M., Christiaens, S., Lee, S.-Y., Andert, J.: Machine-readable specification and intelligent cloud-based execution of logical test cases for automated driving functions. Automated Software Engineering 32, 10 (2025).

[15] Mazkatli, M., Monschein, D., Armbruster, M., Heinrich, R., Koziolek, A.: Continuous integration of architectural performance models with parametric dependencies: the CIPM approach. Automated Software Engineering 32, 54 (2025).

[16] Nguyen, P., Rauniyar, A., Bartel, J., Laufer, J., Dalamagkas, C., Pohl, K.: AI-driven digital twin-based security orchestration, automation and response for critical infrastructures. Automated Software Engineering 33, 61 (2026).

[17] Ghasemi, S., Asgari Araghi, M., Rafe, V., Heckel, R.: MoTDeReL: model-based testing through deep reinforcement learning for software systems specified through graph transformation. Automated Software Engineering 33, 68 (2026).

[18] Ou Ali, M., Agga, M., Ouassaid, M., Maaroufi, M., Elrashidi, A., Kotb, M.F.: A hybrid deep learning model for smart home energy consumption forecasting. Frontiers in Energy Research 12, 1323357 (2024).

[19] Souza, A.B. et al.: Valley-filling electric vehicle charging strategy for demand-side management. PLOS One 20(1), e0316677 (2025).

[20] Zhang, Y., Guan, L., Che, J., Shahidehpour, M.: Deep reinforcement learning for EV charging with safety modules. IEEE Transactions on Smart Grid 15, 757–769 (2024).

[21] Orfanoudakis, S., Diaz-Londono, C., Yılmaz, Y.E., Palensky, P., Vergara, P.P.: EV2Gym: a flexible V2G simulator for EV smart-charging research and benchmarking. IEEE Transactions on Intelligent Transportation Systems 26(2), 2410–2421 (2025).

[22] Li, B., Wei, Y., Li, Z., Dong, Z.Y., Shahidehpour, M.: Detecting false data injection attacks via federated deep learning. IEEE Transactions on Smart Grid 13(6), 4862–4872 (2022).

[23] Han, X., Feng, Y., Li, Y., Zhao, W.: Detecting false data injection attacks via a temporal multi-graph convolutional network. Computers & Security 124, 103016 (2023).

[24] Bitirgen, Ö.F., Filik, Ü.B.: A hybrid deep learning model for discriminating cyber-attacks from disturbances in smart grids. International Journal of Critical Infrastructure Protection 40, 100582 (2023).

[25] Irfan, M. et al.: False data injection attack detection and localisation in smart grids: a review. IET Cyber-Physical Systems: Theory & Applications (2024).

[26] Guo, L., Ye, J., Yang, B.: Cyberattack detection for electric vehicles using physics-guided machine learning. IEEE Transactions on Transportation Electrification 7, 2010–2022 (2021).

[27] Singh, A.R. et al.: AI-based intrusion detection for smart-grid electric-vehicle charging stations. Alexandria Engineering Journal 115, 603–621 (2025).

[28] Qays, M.O., Ahmad, I., Abu-Siada, A., Hossain, M.L., Yasmin, F.: Key communication technologies, applications and challenges of smart grid: a review. Energy Reports 9, 2440–2452 (2023).

[29] Khan, Z.A., Khan, A.N., Waseem, M., Saleh, T.A., Qamar, M.A., Muqeet, A.: Demand response using IoT-enabled architecture in smart grids: a review. IET Renewable Power Generation 18, 3509–3535 (2024).

[30] Homaei, M., Mogollón-Gutiérrez, Ó., Sancho, J., Ávila, M., Caro, A.: A review of digital twins and their application in cybersecurity based on artificial intelligence. Artificial Intelligence Review 57, 201 (2024).

[31] Qureshi, A.R., Asensio, A., Imran, M., Garcia, J., Masip-Bruin, X.: A survey on security enhancing digital twins: models, applications and tools. Computer Communications 238, 108158 (2025).

[32] Suhail, S., Iqbal, M., Hussain, R., Malik, S.U.R., Jurdak, R.: TRIPLE: a blockchain-based digital twin framework for cyber-physical systems security. Journal of Industrial Information Integration 42, 100706 (2024).

[33] Itäpelto, T., Elhajj, M., van Sinderen, M.: Digital twin application in lifecycle security of critical infrastructures: a systematic literature review. International Journal of Critical Infrastructure Protection 50, 100783 (2025).

[34] Moradi, F., Abbaspour Asadollah, S., Pourvatan, B., Moezkarimi, Z., Sirjani, M.: CRYSTAL framework: cybersecurity assurance for cyber-physical systems. Journal of Logical and Algebraic Methods in Programming 139, 100965 (2024).

[35] Alhamam, N., Rahman, M.M.H., Aljughaiman, A.: A comprehensive review on cybersecurity of digital twins: issues, challenges, and future research directions. IEEE Access 13, 45106–45124 (2025).

[36] Villalva, M.G., Gazoli, J.R., Filho, E.R.: Comprehensive approach to modeling and simulation of photovoltaic arrays. IEEE Transactions on Power Electronics 24(5), 1198–1208 (2009).

[37] Tremblay, O., Dessaint, L.-A.: Experimental validation of a battery dynamic model for EV applications. World Electric Vehicle Journal 3(2), 289–298 (2009).

[38] Hannan, M.A., Lipu, M.S.H., Hussain, A., Mohamed, A.: A review of lithium-ion battery state of charge estimation and management system. Renewable and Sustainable Energy Reviews 78, 834–854 (2017).

[39] Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y.: Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Transactions on Smart Grid 10(1), 841–851 (2019).

[40] Wan, Z., Li, H., He, H., Prokhorov, D.: Model-free real-time EV charging scheduling based on deep reinforcement learning. IEEE Transactions on Smart Grid 10(5), 5246–5257 (2019).

[41] Parisio, A., Rikos, E., Glielmo, L.: A model predictive control approach to microgrid operation optimization. IEEE Transactions on Control Systems Technology 22(5), 1813–1827 (2014).

[42] Liu, Y., Ning, P., Reiter, M.K.: False data injection attacks against state estimation in electric power grids. ACM Transactions on Information and System Security 14(1), 1–33 (2011).

[43] Liang, G., Zhao, J., Luo, F., Weller, S.R., Dong, Z.Y.: A review of false data injection attacks against modern power systems. IEEE Transactions on Smart Grid 8(4), 1630–1638 (2017).

[44] Acharya, S., Dvorkin, Y., Pandžić, H., Karri, R.: Cybersecurity of smart electric vehicle charging: a power grid perspective. IEEE Access 8, 214434–214453 (2020).