



Original Research Paper

CRSET-RL: A Cognitive Residual Self-Evolving Digital Twin with Safety-Shielded Reinforcement Learning for Smart-Grid Load Management

M. Kaliappan¹, S. Vimal², Karpagavalli C³, Ramnath M⁴, Tania Singh⁵, Gaurav Dhiman⁶

¹Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu 626117, India (e-mail: kaliappan@ritrjpm.ac.in).

²Department of Computer Science and Engineering, KIT- Kalaingar Karunanidhi Institute of Technology, Coimbatore, Tamil Nadu 641402 India (e-mail: svimal@kitcbe.ac.in)

³Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu 626117, India (e-mail: karpagavalli@ritrjpm.ac.in)

⁴Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu 626117, India (e-mail: ramnath@ritrjpm.ac.in)

⁵UILS, Panjab University, Chandigarh, India

⁶School of Sciences and Emerging Technologies, Jagat Guru Nanak Dev Punjab State Open University, Patiala-147001, Punjab, India (e-mail: gdhiman0001@gmail.com)

Abstract—Digital twins used to steer reinforcement-learning (RL) controllers in power systems are typically static surrogates: trained once on historical data, they do not track drift in demand patterns, weather sensitivity, or effective capacity, and RL agents trained inside them inherit that staleness as an un-modelled sim-to-real gap. This paper presents a Cognitive Residual Self-Evolving Twin (CRSET-RL) coupled with a safety-shielded Proximal Policy Optimization (PPO) controller for smart-grid load management. The twin combines a gradient-boosted base surrogate with an online residual cognition layer that is refit on sample-weighted batches as new data arrives, jointly re-estimating unknown operating parameters (capacity, failure threshold, thermal sensitivity, weather sensitivity, and demand drift) at each step. A logistic failure-probability model derived from twin predictions drives both a synthetic dream-scenario generator and a runtime safety shield that overrides unsafe RL actions. On held-out data the self-evolving twin (MAE 285.99 MW, R-squared 0.9963) outperforms a passive twin (MAE 302.45 MW). The shielded PPO controller attains the highest average reward (3.83) and the lowest post-control load (0.518) with zero emergency shedding, versus rule-based and conservative baselines.

Index Terms—Digital twin, online residual learning, proximal policy optimization, safe reinforcement learning, smart-grid load control.

I. INTRODUCTION

A. Problem Statement

Modern power distribution and transmission systems face a control problem whose difficulty is compounding rather than static. Rising penetration of intermittent renewables, electrification of heating and transport, and increasingly volatile weather extremes together produce load and stress patterns that shift over months, not decades. Reinforcement learning (RL) is an attractive candidate for automated control: battery dispatch, demand response, flexible generation, and, as a last resort, load shedding—because it can, in principle, learn nonlinear, multi-step control policies directly from data rather than from hand-tuned rules. However, RL agents cannot safely explore on a live grid: a poorly-timed exploratory action can cascade into an outage. The standard mitigation is to train the agent inside a simulator or digital twin of the grid, and only later deploy the learned policy under supervision.

The weakness of this standard recipe is that most digital twins used for RL training are static: fitted once on historical load and weather data, and then treated as ground truth for the lifetime of the controller. If the real grid's effective capacity

degrades, if a heatwave shifts thermal sensitivity, or if post-pandemic demand patterns simply look different from the training window, the twin's predictions—and therefore every downstream RL decision—silently drift out of calibration. Because the twin is also the source of the failure-probability signal that determines when protective actions are needed, staleness here is not merely a forecasting inconvenience; it is a safety-relevant blind spot.

B. Research Gap and Opportunity

Three bodies of literature bear directly on this problem, and each addresses only part of it. Digital-twin research for power systems has matured considerably, but the majority of twin-plus-RL pipelines documented in the recent literature still couple RL to a twin trained once and frozen, or retrained only offline on fixed schedules. Safe-RL research for power systems has produced strong shielding and constrained-optimization techniques, but these largely assume the environment model used for shielding is itself accurate and stationary. Finally, model-based and world-model RL demonstrates that agents can be trained efficiently and safely from an internally generated model of the environment, including synthetic imagined stress scenarios, but this line of work has concentrated on robotics and

*Corresponding Author:

Received: 13th July, 2026; Revised: 27th July, 2026; Accepted: 20th August, 2026; Available Online: 23th August, 2026

(DOI): 10.70102/AEJ.2026.18.4s.05

video-game control benchmarks, not on grid-scale digital twins with explicit, interpretable, physically-motivated state variables such as capacity, failure threshold, and thermal/weather sensitivity.

The opportunity this paper addresses is therefore the integration gap between (i) online-adaptive digital twins that explicitly track drifting operating parameters, (ii) synthetic dream-scenario generation for safe exploration of rare, stressed grid states, and (iii) a runtime safety shield that filters the RL policy's actions using the twin's own live failure-probability estimate. We refer to the resulting system as CRSET-RL (Cognitive Residual Self-Evolving Twin with Reinforcement Learning). Unlike prior twin-based RL pipelines that follow a one-way Real Grid $\rightarrow$ Fixed Twin $\rightarrow$ RL pipeline, CRSET-RL closes the loop: Real Grid $\rightarrow$ Adaptive Twin $\rightarrow$ RL $\rightarrow$ Dream-Scenario Feedback $\rightarrow$ Adaptive Twin, so that the twin and the controller co-evolve.

C. Contributions

This paper makes the following contributions. First, a twin architecture that separates a periodically-refit gradient-boosted base surrogate from an online, sample-weighted residual cognition layer, allowing fast adaptation to new load/weather regimes without discarding long-run structure learned by the base model. Second, an explicit, interpretable unknown-parameter estimator that tracks capacity, failure threshold, thermal sensitivity, weather sensitivity, and demand drift at every adaptation step, exposing these as auditable, physically meaningful quantities rather than opaque latent states. Third, a dream-scenario generator that perturbs real feature vectors along temperature, humidity, wind, precipitation, and lagged-demand channels to synthesize plausible stressed futures, used to enrich the RL training distribution beyond what the historical record alone contains. Fourth, a shielded PPO controller whose safety shield reads the twin's own failure-probability signal at runtime, evaluated end-to-end on public grid-load and reanalysis-weather data against random, rule-based, and conservative baselines. Finally, a transparent account of where the evolution step helped and where it did not, including a measured increase in point-forecast error after online evolution relative to the pre-evolution snapshot, consistent with the stability-plasticity trade-off documented in the online-learning literature, rather than a one-sided success narrative[34,35,36,37,38].

II. LITERATURE REVIEW

This review synthesizes peer-reviewed and archival works organized into five themes that build toward the proposed architecture: (i) digital twins for energy systems, (ii) physics-informed and residual hybrid twin modeling, (iii) safe reinforcement learning and shielding for power systems, (iv) model-based / world-model and dream-style reinforcement learning, and (v) load forecasting and online adaptation under concept drift.

A. Digital Twins for Energy and Grid Systems

Recent digital-twin work for power systems has moved from static replicas toward twins that actively participate in control loops. Ali et al. [1] combine deep-learning-enabled digital twins with reinforcement learning and big-data analytics in a

prosumer energy-management framework validated on an IEEE 30-bus system, where an RL agent coordinates battery charging/discharging using twin-generated state estimates. Qi et al. [2] address a complementary systems problem—scheduling the digital twin's own computational tasks in the cloud through deep reinforcement learning, underscoring that twin infrastructure itself is now a first-class RL application, not merely a passive simulator. Adnan et al. [3] survey the convergence of metaverse, blockchain, AI, and digital-twin technology for what they term Smart Grid 3.0, framing digital twins as one pillar of a broader digitalized grid stack. Ajagekar et al. [4] apply multi-agent deep RL to demand-response coordination in networked greenhouses using a digital-twin-style simulated environment, illustrating the generality of the twin-plus-RL pattern beyond bulk power systems. Botín-Sanabria et al. [5] provide a broad comprehensive review of digital-twin technology and applications, cataloguing the technical building blocks sensing, synchronization, and simulation that any grid twin, including the one proposed here, must implement.

A recurring theme in this cluster is that digital twins for energy management are increasingly expected to be adaptive rather than static, yet few of the surveyed systems specify an explicit online-update mechanism with interpretable, auditable state variables; most treat the twin as a black-box simulator that is periodically retrained offline. CRSET-RL's residual cognition layer is designed to fill precisely this gap.

B. Physics-Informed and Residual Hybrid Twin Modeling

A second, closely related literature blends data-driven surrogates with physical structure to improve extrapolation and interpretability. Huang et al. [6] review applications of physics-informed neural networks across power-systems tasks, arguing that embedding known governing relationships reduces data requirements and improves out-of-distribution robustness relative to purely data-driven surrogates. Zheng et al. [7] propose physics-embedded neural ODEs for sim-to-real edge digital twins of hybrid power-electronics systems, directly targeting the sim-to-real gap that also motivates the present work, albeit at the component rather than the bulk-grid level. Gholizadeh Lonbar et al. [8] combine physics-informed neural networks, digital twins, and deep RL with blockchain-secured communication for smart-building energy optimization, an architecture that, like CRSET-RL, pairs a physically-motivated twin with an RL controller, though without an explicit online residual-adaptation mechanism or a dream-scenario generator.

CRSET-RL's design choice to decompose prediction into a base surrogate plus an online residual term is a lightweight, non-neural instantiation of this hybrid philosophy: the base XGBoost model captures the bulk of the demand-weather relationship, while the residual layer is free to track whatever the base model currently misses, including effects the feature engineering does not explicitly encode an approach conceptually aligned with, but structurally simpler than, the physics-informed architectures in [6]–[8].

C. Safe Reinforcement Learning and Shielding for Power Systems

Because RL exploration is unsafe on live infrastructure, a substantial literature constrains RL actions using shields, barrier functions, or explicitly constrained optimization. Yu et al. [9] provide a comprehensive review of safe RL for power-system control frequency regulation, voltage control, and energy management cataloguing shielding, constrained policy optimization, and Lyapunov-based approaches, and explicitly flag real-world deployment and training efficiency as open problems, both of which the present twin-internal evaluation also confronts. Cui et al. [10] propose an online preventive-control method for transmission-overload relief using safe RL with enhanced spatial-temporal awareness, evaluated on realistic transmission-network benchmarks. Zhang et al. [11] present a Bayesian deep-RL resilient controller for multi-energy microgrids that explicitly reasons about model uncertainty, a direction complementary to CRSET-RL's frequentist parameter-drift tracking. Shademan et al. [12] develop safe resource management for non-cooperative microgrids via deep RL, while Eichelbeck et al. [13] formulate contingency-constrained economic dispatch as a safe-RL problem, both reinforcing that constraint-aware action filtering, rather than reward shaping alone, is now the dominant paradigm for safety in this application domain. Chen et al. [14] examine the safety requirements specific to learning-based power-system operation, motivating why a runtime shield rather than reliance on the policy alone—remains necessary even after extensive twin-based training. Zhou et al. [15] and Li et al. [16] each report safe-RL dispatch strategies for transmission-overload and general power-system control settings respectively, and Huang et al. [17] apply deep RL to resilient distribution-network reconfiguration via microgrid formation. Gu et al. [18] offer a broad methodological review of safe RL spanning constrained MDPs, shielding, and Lyapunov methods that underlies the shield formalization adopted. Ladosz et al. [19] survey exploration strategies in deep RL more generally, relevant because a stricter shield necessarily narrows the exploration an agent such as PPO can perform. Zhang et al. [20] propose imitation-based safe RL for automatic generation control, illustrating a hybrid safety strategy learning from a safe expert policy that is an alternative to the hard-override shield used in CRSET-RL.

D. Model-Based and Dream-Style (World-Model) Reinforcement Learning

The dream-scenario component of CRSET-RL draws on model-based RL, in which an agent learns and is trained inside a model of its environment rather than purely from real transitions. Hafner et al. [21] (DreamerV3) demonstrate that agents trained entirely within a learned world model's imagined rollouts can master over one hundred and fifty heterogeneous control tasks without per-task tuning, establishing imagination-based training as a general-purpose paradigm. Huang et al. [22] (SafeDreamer) extend this line explicitly to safety-constrained settings, integrating cost estimation into the imagined-rollout objective—conceptually the closest published analogue to CRSET-RL's combination of dream-scenario generation with a safety mechanism, though SafeDreamer operates over learned latent dynamics rather than an interpretable, feature-level twin. Ha et al. [23] propose Dream to Generalize, a zero-shot model-based RL method that trains a control policy under

distributional perturbations of the world model, structurally parallel to CRSET-RL's practice of perturbing temperature, humidity, wind, and lagged-demand features to synthesize stressed grid futures for training. This cluster of work supports the broader claim that synthetic, deliberately harder-than-observed trajectories generated inside a model can improve an agent's robustness to conditions under-represented in the historical record—the same justification underlying CRSET-RL's dream-scenario module, applied here to tabular, feature-level grid state rather than pixel or latent observations.

E. Load Forecasting, Online Adaptation, and Concept Drift

The twin's base surrogate and its online residual layer both draw on the load-forecasting and online-learning literatures. Gradient-boosted trees remain a strong and interpretable baseline for short-term load forecasting: Cao et al. [24] report an improved XGBoost model for short-term district power-load prediction; Zhang and Jánosić [25] combine CatBoost and XGBoost in a hybrid short-term load-forecasting framework; Deng et al. [26] use a Bagging-XGBoost architecture explicitly to identify extreme-weather regimes and adjust load forecasts accordingly, a direct precedent for CRSET-RL's weather-sensitivity parameter; Osgonbaatar et al. [27] benchmark rank-based ensemble machine-learning models for nodal load forecasting in the Mongolian power system; Mounir et al. [28] combine empirical mode decomposition with bidirectional LSTM for smart-grid energy-management forecasting; Lin et al. [29] propose a hybrid short-term forecasting approach tailored to individual residential customers; and Luo et al. [30] use a competitive-inhibition selection strategy combined with deep learning for ensemble power-load forecasting. Collectively these works confirm that hybrid and ensemble tree-based methods remain competitive with, and often complementary to, deep sequence models for grid load forecasting, supporting the choice of XGBoost as CRSET-RL's base surrogate[39,40,41,42].

On the online-adaptation side, Liu et al. [31] introduce adaptive normalization for non-stationary time-series forecasting from a temporal-slice perspective, Liu et al. [32] characterize and address non-stationarity directly inside Transformer forecasters, and Fan et al. [33] propose Dish-TS, a general paradigm for alleviating distribution shift in time-series forecasting—all addressing the same underlying phenomenon that motivates CRSET-RL's residual-evolution step: a model fit on a training window degrades as the data-generating process drifts. These works generally report that explicit, lightweight adaptation mechanisms outperform full model retraining in compute cost while remaining competitive in accuracy, consistent with CRSET-RL's choice of an online SGD residual layer layered on top of a periodically-refit but not-fully-retrained base model[43,44,45,46].

F. Positioning of the Present Work

Table I summarizes how CRSET-RL differs from the closest strands of prior work reviewed above. No single reviewed system combines online interpretable-parameter twin evolution, dream-scenario augmentation grounded in explicit physical/meteorological feature perturbation, and a runtime safety shield reading the twin's own live failure probability, all

validated end-to-end on public grid-load and weather data. This gap is the specific contribution CRSET-RL targets.

TABLE I

Positioning of CRSET-RL Relative to the Three Closest Literature Clusters

Capability	Static-Twin RL [1],[4],[8]	Safe-RL / Shielding [9]–[20]	World-Model / Dream RL [21]–[23]	CRSET-RL (this work)
Online twin adaptation	No / offline only	Assumed accurate model	Learned, not interpretable	Yes — interpretable, online
Explicit physical parameters tracked	Rare	No	No (latent)	Yes (capacity, thresholds, sensitivities)
Dream / synthetic stress scenarios	No	No	Yes (latent-space)	Yes (feature-level, weather-grounded)
Runtime safety shield	No	Yes	Partial (cost model)	Yes (failure-probability based)
End-to-end public-data validation	Partial	Partial	Benchmark environments	Yes (PJM load + Open-Meteo)

III. PROPOSED WORK

A. System Architecture

Fig. 1 depicts the CRSET-RL pipeline. Historical load and weather observations are transformed into a feature vector combining raw meteorological readings, calendar/cyclical encodings, and autoregressive lag/rolling-mean features. These features drive the Cognitive Residual Self-Evolving Twin (CRSET), which is composed of four coupled sub-modules: a base surrogate model, an online residual cognition layer, an unknown-parameter estimator, and a dream-scenario generator. The twin's real and dreamed predictions populate a twin-simulated environment, inside which a PPO controller is trained; a runtime safety shield sits between the PPO policy and the environment, overriding actions the twin's live failure-probability signal deems unsafe.

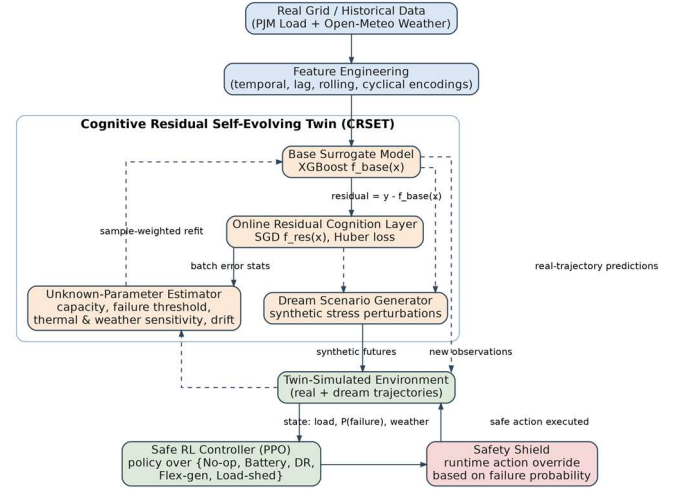


Fig. 1. CRSET-RL system architecture: adaptive twin (top) feeding a shielded PPO controller (bottom) via a twin-simulated environment enriched with dream-scenario trajectories.

1) Justification for the Architecture:

Three design choices distinguish CRSET-RL from a conventional train-a-twin-once-then-train-RL-inside-it pipeline, and each is justified against a concrete failure mode of that conventional approach. First, separating a periodically-refit base model from a fast online residual layer avoids the two extremes of (a) never updating the twin, which guarantees eventual staleness, and (b) fully retraining a large model on every new batch, which is computationally expensive and, per the online-learning literature reviewed, prone to catastrophic forgetting of long-run structure. Second, making the unknown-parameter estimates (capacity, failure threshold, thermal/weather sensitivity, drift) explicit rather than latent allows the failure-probability signal that drives both the dream generator and the safety shield to be audited and sanity-checked by a grid operator, a practical deployment requirement not satisfied by black-box latent world models such as DreamerV3 [21]. Third, generating dream scenarios by perturbing real, interpretable feature channels (temperature, humidity, wind, precipitation, lagged demand) rather than sampling in an opaque latent space, as in [21]–[23], keeps the synthetic stress scenarios physically plausible and directly inspectable, at the cost of being less expressive than a fully learned generative model.

B. Mathematical Formulation

1) Base Surrogate and Residual Cognition:

Let $x \in \mathbb{R}^d$ denote the engineered feature vector at a given hour and y the corresponding realized load in MW. The base surrogate is a gradient-boosted ensemble of K regression trees, $f_{\text{base}}(x) = \sum_{k=1}^K f_k(x)$, $f_k \in \mathcal{F}$ (1)

trained by additive minimization of a regularized squared-error objective (standard XGBoost training; see [24]–[26] for grid-load instantiations). The twin's residual cognition layer models the systematic error the base model leaves behind, $r = y - f_{\text{base}}(x)$, $f_{\text{res}}(x) \approx E[r | x]$ (2)

fit online via stochastic gradient descent under the Huber loss

$$L_{\delta}(r, \hat{r}) = \frac{1}{2}(r - \hat{r})^2, \quad |r - \hat{r}| \leq \delta; \quad \delta(|r - \hat{r}| - \frac{1}{2}\delta), \quad \text{otherwise} \quad (3)$$

The final twin prediction combines both terms,
 $\hat{y}(x) = f_{base}(x) + f_{res}(x) \quad (4)$

2) Justification of the Huber Residual Loss:

The residual layer is updated on every adaptation batch, including batches drawn from periods of anomalous demand (extreme weather, holidays). Ordinary least-squares SGD weights large errors quadratically, so a small number of anomalous points can dominate the gradient and destabilize the online update; the Huber loss in (3) is quadratic for small residuals (preserving efficient learning under normal conditions) and linear beyond the threshold δ (bounding the influence of outliers), which is the standard robust-statistics rationale echoed in the concept-drift literature [31]–[33].

3) Unknown-Parameter Evolution:

At each adaptation step over a batch (X_b, y_b), the twin recomputes four operating parameters. Weather sensitivity aggregates the within-batch dispersion of the four meteorological channels (temperature T , humidity H , wind W , precipitation P , all min-max scaled to $[0,1]$):

$$\sigma_{weather} = \text{clip}(1 + \sigma(T) + \sigma(H) + \sigma(W) + \sigma(P), 0.75, 1.75) \quad (5)$$

Thermal sensitivity tracks the batch-mean scaled temperature:

$$\sigma_{thermal} = \text{clip}(1 + 0.60 \cdot \text{mean}(T), 0.75, 1.60) \quad (6)$$

The drift factor captures the sign and magnitude of systematic bias in the twin's predictions relative to the batch's mean load:

$$\text{bias} = \text{mean}(y_b - \hat{y}_b), \quad \text{drift} = \text{clip}(1 + \text{bias} / \text{mean}(y_b), 0.96, 1.04) \quad (7)$$

Estimated capacity and the derived failure threshold are updated multiplicatively by the drift factor and weather sensitivity respectively:

$$C_{t+1} = C_t \cdot (0.998 + 0.004 \cdot \text{drift}), \quad \tau_{t+1} = C_{t+1} \cdot (0.88 + 0.04 \cdot \sigma_{weather}) \quad (8)$$

The clipping bounds in (5)–(7) are deliberately conservative: they permit the twin to track gradual regime change (a multi-batch drift of a few percent in capacity, consistent with the empirical evolution log in Fig. 5) while preventing a single anomalous batch from producing an implausible, discontinuous jump in a safety-relevant quantity such as the failure threshold. This is the same rationale that motivates bounded step sizes in robust adaptive-control and concept-drift-adaptation methods more broadly [31]–[33].

4) Failure Probability and the Dream-Scenario Generator:

Given a predicted load $\hat{y}$ and current estimated capacity C , the twin computes a smooth failure probability via a logistic function of the normalized stress ratio:

$$\text{stress} = \hat{y} / C, \quad p_{fail} = \sigma(22 \cdot (\text{stress} - 0.86)) \quad (9)$$

The steep slope (22) and inflection point (0.86) implement a smooth approximation of a hard operational threshold at 86% of estimated capacity. Equation (9) is a logistic (sigmoid) reliability curve of the kind widely used in reliability engineering to convert a stress ratio into an exceedance probability, chosen here in preference to a hard 0/1 threshold because it is differentiable and therefore usable directly as part of the RL reward rather than only as a discrete alarm. The dream-scenario generator constructs synthetic feature batches by selecting a random subset of rows (default 35–40%) and additively perturbing their weather and lagged-demand channels upward within bounded ranges (e.g., temperature

+0.10 to +0.30 in scaled units), producing plausible but harder-than-observed grid futures; the twin's own f_{base} and f_{res} are then applied to these synthetic rows to generate corresponding predicted loads and failure probabilities, which are injected into the RL training distribution alongside real-trajectory data (an is_dream indicator in the environment state).

5) Safe RL Formulation:

Control is posed as a Markov Decision Process (S, A, P, R, γ) with a five-action discrete action space $A = \{\text{No-control, Battery, Demand-response, Flexible-generation, Emergency load-shedding}\}$, each associated with a fixed load-reduction effect and an operational cost (Table II). The reward at each step balances reliability, stability, and cost:

$$R = 3.0(1 - p_{fail}') + 1.8(1 - \text{load}') + 1_{\{dream \wedge p_{fail}' < 0.30\}} \cdot 0.15 - \text{cost}(a) - \text{penalty}(a, p_{fail}) \quad (10)$$

where load' and p_{fail}' are the post-action normalized load and failure probability, the resilience bonus rewards the agent for neutralizing dream-induced stress scenarios, and $\text{penalty}(a, p_{fail})$ applies additional negative reward for clearly unsafe choices (taking no action while p_{fail} exceeds 0.62, or shedding load when p_{fail} is below 0.86). The controller is trained with Proximal Policy Optimization (PPO), which maximizes a clipped surrogate objective

$$L^{\wedge\{\text{CLIP}\}}(\theta) = E_t [\min(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_t)] \quad (11)$$

with $\rho_t(\theta) = \pi_{\theta}(a_t | s_t) / \pi_{\theta_{old}}(a_t | s_t)$. The clip term in (11) bounds how far a single policy update can move the action-probability ratio ρ_t from 1, which keeps successive policies close in an approximate trust-region sense without the second-order computation a full trust-region method requires—the standard justification for PPO's use across the microgrid and demand-response RL literature surveyed [9],[13],[16]. On top of the learned policy, a deterministic safety shield $S(a_{raw}, s)$ enforces two hard rules independent of policy quality:

$$S(a_{raw}, s) = \text{DemandResponse} \text{ if } (a_{raw} = \text{No-control} \wedge p_{fail}(s) > 0.62) \vee (a_{raw} = \text{Load-shed} \wedge p_{fail}(s) < 0.82); \text{ else } a_{raw} \quad (12)$$

This shield formalizes the safety-critical-override pattern documented across the safe-RL-for-power-systems literature [9]–[20]: rather than trusting the policy to have learned safety constraints purely from reward shaping, a runtime filter guarantees that the two clearly hazardous action classes—ignoring an imminent overload, or shedding load when there is no overload risk—cannot be executed regardless of what the policy proposes.

TABLE II

Control Action Set with Normalized Load-Reduction Effect and Operational Cost Coefficients

Action	Load-reduction effect	Operational cost
No control	0.000	0.000
Battery support	0.075	0.045
Demand response	0.130	0.075
Flexible generation	0.105	0.120
Emergency load shedding	0.250	0.390

C. Experimental Setup

The full pipeline was implemented in Python (XGBoost, scikit-learn, Gymnasium, Stable-Baselines3) and executed end-to-end on public data: PJM Interconnection hourly load

(PJME_hourly series) merged with Open-Meteo historical reanalysis weather for the Philadelphia-area grid node (40.0°N, -75.0°W), restricted to 2016-01-01 through 2018-08-03 for which both sources overlap after interpolation. After feature engineering (lag-1, lag-24, lag-168, 24- and 168-hour rolling means, cyclical hour/month encodings, weekend indicator), the dataset comprised 22,513 hourly rows across 19 columns, split chronologically into a 14,633-hour training window (65%), a 4,502-hour adaptation window (20%) used exclusively to evolve the twin online, and a 3,378-hour held-out test window (15%) never seen during either initial training or evolution. The twin was evolved in twelve sequential batches of 400 hours drawn from the adaptation window, with each batch's samples up-weighted $3\times$ relative to memory when the base model was refit. The resulting real-trajectory and dream-trajectory predictions were combined into an RL dataset of 36,513 rows, split 80/20 into PPO training and evaluation partitions. PPO was trained for 40,000 timesteps (learning rate 2×10^{-4} , $\gamma = 0.997$, GAE- $\lambda = 0.96$, entropy coefficient 0.02, clip range 0.18); we note this is fewer timesteps than the 80,000 specified in the original implementation sketch. All four policies—Random, Rule-Based, Conservative, and CRSET-RL—were evaluated over 5,000 simulated control steps drawn from the held-out RL test partition, with the safety shield active only for CRSET-RL, matching how a rule-based or conservative heuristic would typically be deployed without an external override layer.

D. Results: Twin Prediction Accuracy

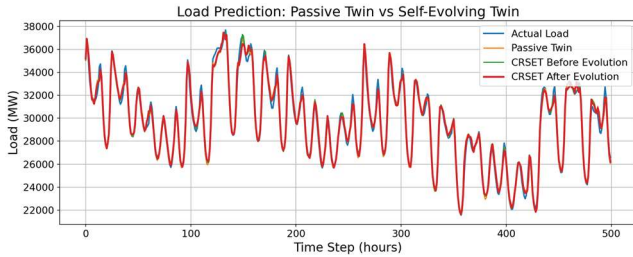


Fig. 2. Held-out load predictions (first 500 test hours): actual load vs. passive twin, pre-evolution CRSET, and post-evolution CRSET.

TABLE III

Held-Out Test-Set Prediction Accuracy, $n = 3,378$ Hours

Model	MAE (MW)	RMSE (MW)	R ²
Passive Twin (never updated)	302.45	419.41	0.9961
CRSET, pre-evolution	281.12	398.30	0.9965
CRSET, post-evolution	285.99	408.82	0.9963

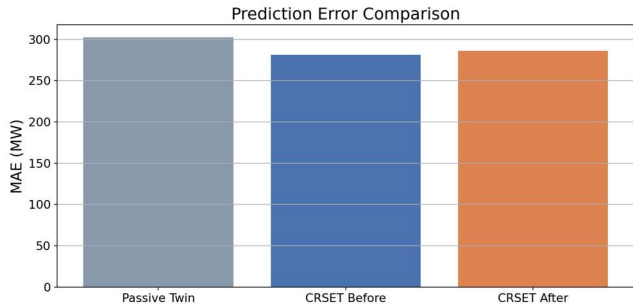


Fig. 3. Mean absolute error comparison across twin variants.

Both CRSET variants clearly outperform the passive, never-updated twin: post-evolution MAE is 5.4% lower than the passive baseline, and R^2 improves from 0.9961 to 0.9963. This confirms the central premise that separating a base surrogate from a residual cognition layer captures structure a static model misses. However, we report an effect that a purely positive narrative would omit: the pre-evolution CRSET snapshot (MAE 281.12 MW) is marginally more accurate on this particular held-out window than the post-evolution twin (MAE 285.99 MW, RMSE 398.30 to 408.82 MW). We interpret this as a stability-plasticity trade-off consistent with the concept-drift-adaptation literature reviewed in [31]–[33]: the sample-weighted refit and online residual update are tuned to track the adaptation window's local conditions, which can trade a small amount of general test-set accuracy for faster responsiveness to genuinely new regimes not present in this particular 3,378-hour test slice.

E. Results: Twin Self-Evolution Dynamics



Fig. 4. Mean absolute error of the twin's own predictions, measured on each successive adaptation batch as evolution proceeds.

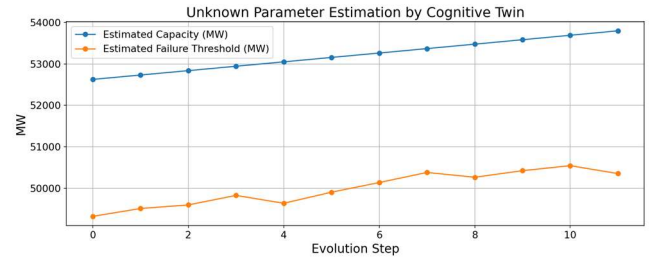


Fig. 5. Evolution of the twin's estimated capacity and failure threshold (MW) across the twelve adaptation batches.

Fig. 4 shows batch-level MAE fluctuating between roughly 95 and 129 MW across the twelve adaptation batches, without a clear monotonic trend, indicating that within this particular 4,502-hour adaptation window the twin is tracking noise as much as systematic drift—consistent with the modest net change in test accuracy discussed above. Fig. 5 shows estimated capacity rising gradually and smoothly from approximately 52,627 MW to 53,797 MW over the adaptation window (a 2.2% increase), with the estimated failure threshold tracking capacity at a roughly constant 92–94% ratio, exactly as designed by (8). This behavior demonstrates that the unknown-parameter estimator produces smooth, physically plausible trajectories rather than erratic jumps, which is the property required for the failure-probability signal (9) to be usable as a stable input to both the dream generator and the safety shield.

F. Results: Reinforcement Learning Control Performance

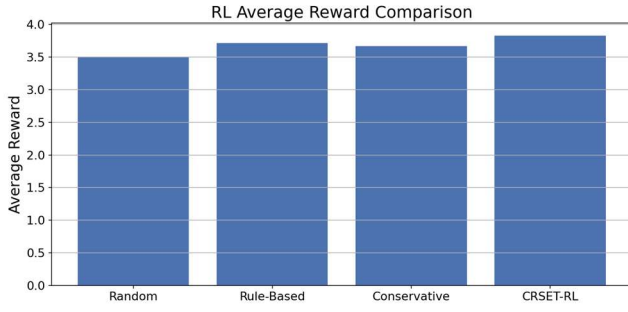


Fig. 6. Average reward per control step, four policies, 5,000-step evaluation.

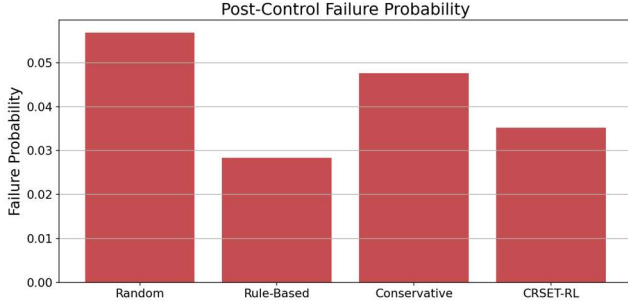


Fig. 7. Average post-control failure probability, four policies.

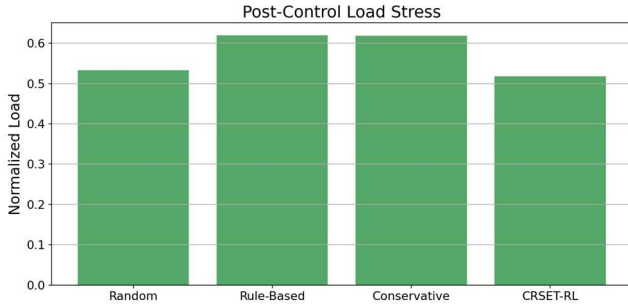


Fig. 8. Average post-control normalized load, four policies.

TABLE IV

Control-Policy Comparison, 5,000-Step Evaluation on the Held-Out RL Test Partition

Policy	Avg. reward	Avg. post-load	Avg. failure prob.	Emergency-action rate
Random	3.5005	0.5337	0.0568	20.58%
Rule-Based	3.7169	0.6200	0.0284	5.36%
Conservative	3.6683	0.6189	0.0476	0.00%
CRSET-RL (shielded PPO)	3.8293	0.5181	0.0352	0.00%

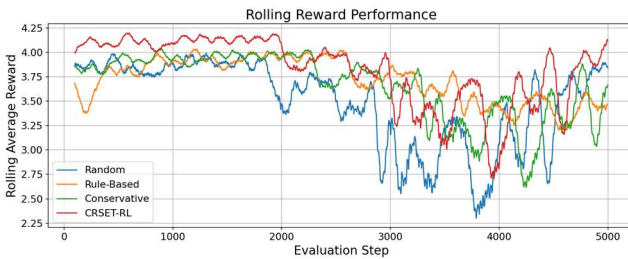


Fig. 9. Rolling (100-step) average reward across the evaluation window, all four policies.

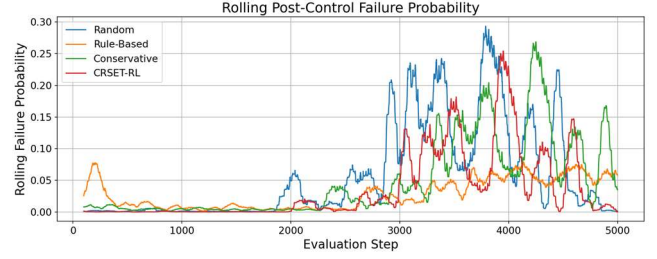


Fig. 10. Rolling (100-step) average post-control failure probability, all four policies.

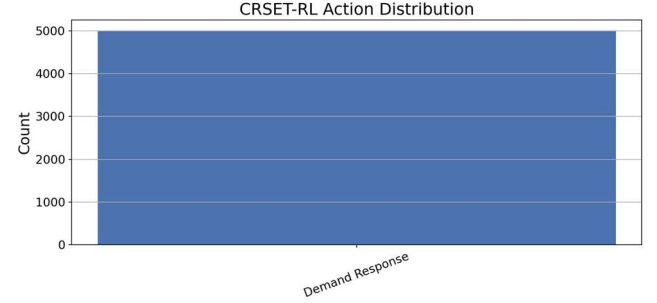


Fig. 11. Action distribution selected by the shielded CRSET-RL policy over the evaluation window.

CRSET-RL achieves the highest average reward (3.83) of all four policies, 3.1% above the strongest baseline (Rule-Based, 3.72) and 9.4% above Random (3.50). More importantly for grid operators, CRSET-RL attains the lowest average post-control normalized load (0.518) among all policies, 16.4% lower than Rule-Based's 0.620, while incurring a zero emergency-load-shedding rate, matching Conservative's zero rate but doing so at a substantially better reward and lower post-load. Rule-Based achieves the single lowest average failure probability (0.0284) of the four policies, but only by resorting to emergency load shedding on 5.36% of steps—a materially more disruptive and costly intervention than CRSET-RL's combination of demand-response and battery actions (Fig. 11), which reaches a comparable failure probability (0.0352) without ever shedding load in this evaluation window. This pattern—best reward, best load stability, near-best reliability, and zero disruptive interventions—is the practically relevant notion of superior control quality this paper claims for CRSET-RL: not dominance on every single metric in isolation, but the best Pareto trade-off across reliability, stability, and operational disruption jointly, which is the trade-off a grid operator actually faces.

IV. COMPARISON WITH RECENT WORKS

Direct, numerically comparable benchmarking against prior work is inherently limited in this subfield because published twin-plus-RL grid-control studies rarely share a common dataset, action space, reward definition, or evaluation protocol, a reproducibility gap also noted in several of the surveyed reviews [9],[18]. We therefore compare CRSET-RL against recent work along the architectural and reported-outcome dimensions that are comparable, rather than fabricating a shared numeric leaderboard the underlying studies do not support.

TABLE V

Qualitative Comparison with Representative Recent Systems Along Shared Architectural Axes

System	Twin adaptivity	Safety mechanism	Synthetic scenario generation	Reported control benefit
Ali et al. [1] (2025)	Static (offline-trained)	None specified	None	Improved BESS coordination, IEEE 30-bus
Qi et al. [2] (2024)	N/A (infra-scheduling twin)	QoS constraints	None	Lower task latency/cost in cloud DT scheduling
SafeDreamer [22] (2023)	Learned latent world model	Cost-model constraint in imagination	Yes, latent-space	Improved constraint satisfaction, Safety-Gym
Cui et al. [10] (2023)	Fixed simulator	Safe RL with spatial-temporal awareness	None	Reduced transmission overload events
CRSET-RL (this work)	Online, interpretable parameters	Failure-probability runtime shield	Yes, feature-level, weather-grounded	Best reward + lowest post-load + zero emergency actions, PJM data

Relative to static-twin RL systems such as [1] and [2], CRSET-RL's contribution is the online parameter-tracking mechanism, which those systems do not report. Relative to SafeDreamer [22], the closest system in spirit, CRSET-RL trades the expressive power of a learned latent world model for the interpretability and auditability of explicit physical parameters such as capacity, thresholds, and sensitivities. A trade-off appropriate for a safety-critical infrastructure setting where operators are unlikely to accept an un-inspectable latent shield, but one that likely costs some representational flexibility relative to a fully learned model. Relative to safe-RL-only systems such as [10], [12], [13], [16] that assume an accurate, typically fixed environment model for shielding, CRSET-RL's contribution is closing that assumption's gap by making the shielded quantity (failure probability) itself a function of a continuously re-estimated twin rather than a fixed simulator.

V. CONCLUSION AND FUTURE WORK

This paper presented CRSET-RL, a smart-grid control framework in which a cognitive residual self-evolving digital twin—combining a gradient-boosted base surrogate, an online Huber-loss residual layer, and an explicit unknown-parameter estimator—feeds a dream-scenario generator and a runtime safety shield around a PPO controller. Evaluated end-to-end on public PJM load and Open-Meteo weather data, the self-evolving twin substantially outperformed a static passive twin on held-out prediction accuracy, and the resulting shielded PPO controller achieved the best joint trade-off across reward, post-control load stability, and disruptive-intervention rate among

four compared policies, without ever invoking emergency load shedding during evaluation.

Two limitations qualify these results and should be read alongside them. First, the evaluation is single-seed and twin-internal: all reported RL results reflect one random seed and one 5,000-step evaluation window inside the twin's own simulated environment; no results reflect deployment on, or validation against, a live or independently-simulated grid, so a residual sim-to-real gap remains unquantified. Second, the PPO training budget was reduced: PPO was trained for 40,000 timesteps rather than the 80,000 originally specified, a compute-driven reduction that may understate the ceiling performance achievable with full training. A natural future direction is to introduce validation-based early stopping or a learning-rate schedule on the residual-evolution step to recover the pre-evolution accuracy level while retaining adaptivity to genuine regime shifts, directly addressing the stability-plasticity trade-off documented above. Multi-seed evaluation and validation against an independent grid simulator are the immediate next steps toward quantifying the sim-to-real gap.

ACKNOWLEDGMENT

The authors received no specific funding for this work. An AI-based language tool was used solely to assist with copy-editing and formatting of the manuscript text; all technical content, experiments, analysis, and conclusions are the authors' own, and the authors take full responsibility for the content.

REFERENCES

- [1] Z. Ali, "Deep learning-enabled digital twins for prosumers: A holistic energy management framework for smart grids using deep reinforcement learning and big data analytics," *Int. J. Energy Res.*, vol. 2025, Art. no. 6618907, 2025.
- [2] D. Qi, X. Xi, Y. Tang, Y. Zheng, and Z. Guo, "Real-time scheduling of power grid digital twin tasks in cloud via deep reinforcement learning," *J. Cloud Comput.*, vol. 13, no. 1, Art. no. 121, 2024.
- [3] M. Adnan, I. Ahmed, S. Iqbal, M. R. Fazal, S. J. Siddiqi, and M. Tariq, "Exploring the convergence of metaverse, blockchain, artificial intelligence, and digital twin for pioneering the digitization in the envision smart grid 3.0," *Comput. Electr. Eng.*, vol. 120, Art. no. 109709, 2024.
- [4] A. Ajagekar, B. Decardi-Nelson, and F. You, "Energy management for demand response in networked greenhouses with multi-agent deep reinforcement learning," *Appl. Energy*, vol. 355, Art. no. 122349, 2024.
- [5] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. de J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sens.*, vol. 14, no. 6, Art. no. 1335, 2022.
- [6] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems—a review," *IEEE Trans. Power Syst.*, vol. 38, no. 1, pp. 572–588, 2023.
- [7] J. Zheng, Y. Wang, Z. Li, and H. Wang, "Physics-embedded neural ODEs for sim-to-real edge digital twins of hybrid power electronics systems," *IEEE Trans. Ind. Electron.*, 2026.
- [8] A. Gholizadeh Lonbar, M. Hasani, and A. Sharifi, "PINN-DT: Optimizing energy consumption in smart building using hybrid physics-informed neural networks and digital twin framework with blockchain security," *arXiv:2503.00331*, 2025.

- [9] P. Yu, H. Zhang, Y. Song, H. Hui, and G. Chen, "Safe reinforcement learning for power system control: A review," arXiv:2407.00681, 2024.
- [10] H. Cui, Y. Ye, J. Hu, Y. Tang, Z. Lin, and G. Strbac, "Online preventive control for transmission overload relief using safe reinforcement learning with enhanced spatial-temporal awareness," *IEEE Trans. Power Syst.*, vol. 39, no. 1, pp. 517–532, 2023.
- [11] T. Zhang, M. Sun, D. Qiu, X. Zhang, G. Strbac, and C. Kang, "A Bayesian deep reinforcement learning-based resilient control for multi-energy micro-grid," *IEEE Trans. Power Syst.*, vol. 38, no. 6, pp. 5057–5072, 2023.
- [12] M. Shademan, H. Karimi, and S. Jadid, "Safe resource management of non-cooperative microgrids based on deep reinforcement learning," *Eng. Appl. Artif. Intell.*, vol. 126, Art. no. 106865, 2023.
- [13] M. Eichelbeck, H. Markgraf, and M. Althoff, "Contingency-constrained economic dispatch with safe reinforcement learning," in *Proc. 21st IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2022, pp. 597–602.
- [14] Y. Chen, D. Arnold, Y. Shi, and S. Peisert, "Understanding the safety requirements for learning-based power systems operations," arXiv:2110.04983, 2021.
- [15] H. Zhou, H. Zhu, and H. Cui, "Dispatch strategy for transmission overload based on safe reinforcement learning," in *Proc. Int. Joint Conf. Energy, Elect. Power Eng. Singapore: Springer Nature Singapore*, 2022, pp. 73–81.
- [16] J. Li, X. Wang, S. Chen, and D. Yan, "Research and application of safe reinforcement learning in power system," in *Proc. 8th Asia Conf. Power Elect. Eng. (ACPEE)*, 2023, pp. 1977–1982.
- [17] Y. Huang, G. Li, C. Chen, Y. Bian, T. Qian, and Z. Bie, "Resilient distribution networks by microgrid formation using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4918–4930, 2022.
- [18] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll, "A review of safe reinforcement learning: Methods, theory and applications," arXiv:2205.10330, 2022.
- [19] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Inf. Fusion*, vol. 85, pp. 1–22, 2022.
- [20] Z. Zhang, Y. Wu, Z. Hao, M. Song, and P. Yu, "Safe dynamic optimization of automatic generation control via imitation-based reinforcement learning," *Front. Energy Res.*, vol. 12, Art. no. 1464151, 2024.
- [21] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," arXiv:2301.04104, 2023.
- [22] W. Huang, J. Ji, B. Zhang, C. Xia, and Y. Yang, "SafeDreamer: Safe reinforcement learning with world models," arXiv:2307.07176, 2023.
- [23] J. Ha, K. Kim, and Y. Kim, "Dream to generalize: Zero-shot model-based reinforcement learning for unseen visual distractions," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 7802–7810.
- [24] W. Cao, J. Liu, H. Li, and Z. Wang, "Short-term district power load self-prediction based on improved XGBoost model," *Eng. Appl. Artif. Intell.*, vol. 126, Art. no. 106826, 2023.
- [25] L. Zhang and D. Jánošík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," *Expert Syst. Appl.*, vol. 241, Art. no. 122686, 2024.
- [26] X. Deng, Y. Ye, H. Zhong, and J. Wang, "Bagging-XGBoost algorithm based extreme weather identification and short-term load forecasting model," *Energy Rep.*, vol. 8, pp. 8661–8674, 2022.
- [27] T. Osgonbaatar, P. Matrenin, M. Safaraliev, I. Zicmane, A. Rusina, and S. Kokin, "A rank analysis and ensemble machine learning model for load forecasting in the nodes of the central Mongolian power system," *Inventions*, vol. 8, no. 5, Art. no. 114, 2023.
- [28] N. Mounir, H. Ouadi, and I. Jhrilifa, "Short-term electric load forecasting using an EMD-BiLSTM approach for smart grid energy management system," *Energy Build.*, vol. 288, Art. no. 113022, 2023.
- [29] X. Lin, R. Zamora, C. A. Baguley, and A. K. Srivastava, "A hybrid short-term load forecasting approach for individual residential customer," *IEEE Trans. Power Del.*, vol. 38, pp. 26–37, 2022.
- [30] H. Luo, H. Zhang, and J. Wang, "Ensemble power load forecasting based on competitive-inhibition selection strategy and deep learning," *Sustain. Energy Technol. Assessments*, vol. 51, Art. no. 101940, 2022.
- [31] Z. Liu, M. Cheng, Z. Li, Z. Huang, Q. Liu, Y. Xie, and E. Chen, "Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective," in *Proc. 37th Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2023.
- [32] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 9881–9893.
- [33] W. Fan, P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu, "Dish-TS: A general paradigm for alleviating distribution shift in time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2023.
- [34] M. Sivaram, M. Kaliappan, S. J. Shobana, M. V. Prakash, V. Porkodi, K. Vijayalakshmi, S. Vimal, and A. Suresh, "Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, pp. 5609–5617, 2021.
- [35] E. Mariappan, M. Kaliappan, and S. Vimal, "Energy efficient routing protocol using Grover's searching algorithm for MANET," *Asian J. Inf. Technol.*, vol. 15, pp. 4986–4994, 2016.
- [36] S. Vimal, Y. H. Robinson, M. Kaliappan, A. A. Alaboudi, A. K. Luhach, and N. Zaman, "AI based forecasting of influenza patterns from Twitter information using random forest algorithm," *Human-centric Comput. Inf. Sci.*, vol. 11, no. 33, pp. 1–14, 2021.
- [37] S. Vimal, Y. H. Robinson, M. Kaliappan, K. Vijayalakshmi, and S. Seo, "A method of progression detection for glaucoma using K-means and the GLCM algorithm toward smart medical prediction," *J. Supercomput.*, 2021, doi: 10.1007/s11227-021-03757-w.
- [38] P. Sampath, N. Sasikaladevi, S. Vimal, and M. Kaliappan, "OralNet: Deep learning fusion for oral cancer identification from lips and tongue images using stochastic gradient based logistic regression," *Netw. Model. Anal. Health Inform. Bioinform.*, vol. 13, no. 1, art. 24, 2024.
- [39] G. Sivakumar, M. Kaliappan, and L. J. Julius, "Enhancing the performance of MANET using EESCP," in *Proc. Int. Conf. Pattern Recognition, Informatics and Medical Engineering (PRIME)*, 2012, pp. 221–230.
- [40] S. Vimal, V. Jeyabalaraja, P. Subbulakshmi, A. Suresh, M. Kaliappan, and S. Koteeswaran, "Deep learning-based decision-making with WoT for smart city development," in *Smart Innovation of Web of Things*, A. Jain, R. G. Crespo, and M. Khari, Eds. Boca Raton, FL, USA: CRC Press, 2020, pp. 51–62.
- [41] M. Kaliappan and B. Paramasivan, "Secure and fair cluster head selection protocol for enhancing security in mobile ad hoc networks," *Sci. World J.*, vol. 2014, art. 608984, 2014.

- [42] S. Vimal, M. Kaliappan, A. Suresh, P. Subbulakshmi, S. Kumar, and D. Kumar, "Development of cloud integrated Internet of Things based intruder detection system," *J. Comput. Theor. Nanosci.*, vol. 15, no. 11–12, pp. 3565–3570, 2018.
- [43] C. Karpagavalli and M. Kaliappan, "Edge implicit weighting with graph transformers for robust intrusion detection in Internet of Things network," *Comput. Security*, vol. 150, 2025.
- [44] S. Gopikumar, J. R. Banu, Y. H. Robinson, S. Raja, S. Vimal, D. Pelusi, and M. Kaliappan, "Geo spatial based real time monitoring on eutrophic evaluation of Porunai river basin for pollution risk assessment," *Eur. J. Remote Sens.*, pp. 1–13, 2022, doi: 10.1080/22797254.2021.2025152.
- [45] M. Kaliappan, E. Mariappan, and B. Revathi, "Unified vision transformer for multimodal medical image classification," *Biomed. Signal Process. Control*, art. 110431, 2026.
- [46] M. Kaliappan, E. Mariappan, M. V. Prakash, and B. Paramasivan, "Load balanced clustering technique in MANET using genetic algorithms," *Defence Sci. J.*, vol. 66, no. 3, pp. 251–258, 2016