Uncertainty Quantification and Failure Prediction for AI-based Congestion Management

Margarida Costa, Pedro G. Ferreira
Laboratory of Artificial Intelligence and Decision Support (LIAAD)
INESC TEC and FCUP
Porto, Portugal
{margarida.a.costa, pedro.g.ferreira}@inesctec.pt

Carla Gonçalves, Ricardo J. Bessa
Center for Power and Energy Systems (CPES)
INESC TEC and FEUP
Porto, Portugal
{carla.s.goncalves, ricardo.j.bessa}@inesctec.pt

***Abstract*—The increasing integration of renewable energy sources and demand electrification is increasing the operational complexity of power grids. Reinforcement learning (RL) agents have shown promising performance in real-time congestion management with topology optimization, but quantifying their reliability under unseen operating conditions remains difficult. This paper proposes a framework to quantify and predict the reliability of pre-trained RL agents for real-time congestion management. The approach integrates epistemic and aleatoric uncertainties into a machine learning–based failure prediction model that estimates the probability of RL agent failure one-hour-ahead, accounting for potential contingency scenarios. Results in the *Grid2Op* environment demonstrate the proposed framework's ability to anticipate unreliable AI recommendations, achieving an accuracy of 89.5% while maintaining an oversight rate below 7% and a false alarm rate below 11%.**

***Index Terms*—Uncertainty, reinforcement learning, congestion, interpretability, failure prediction, large language models**

I. Introduction

The operation of modern power grids is becoming increasingly complex due to the large-scale integration of renewable energy sources (RES) and emerging high-demand loads such as data centers and electric vehicle fast-charging stations. These developments increase variability in power injections and the need for real-time operational decisions, including congestion and contingency management. Consequently, AI-based operational assistants with fast inference capabilities are gaining attention [1]. Reinforcement learning (RL) has shown promising results for topology control in simulation [2], but its behavior under rare or unseen operating conditions remains difficult to assess [3]. Beyond average performance, operators must know when an RL agent's recommendations become unreliable, for example, when encountering unfamiliar grid states or forecast-driven variability in generation and demand. Without uncertainty quantification, operators may overtrust AI recommendations. AI systems should therefore exhibit *meta-awareness* [4], recognizing when situations exceed their capabilities and prompting human intervention. In this context, this paper proposes an uncertainty-aware prediction framework to anticipate operating conditions under which AI assistants may generate low-confidence recommendations for real-time congestion management.

This research is being carried out as a part of the AI4REALNET (European Union's Horizon Europe, No. 101119527) project. The sole responsibility of this publication lies with the authors. The European Union is not responsible for any use that may be made of the information contained therein.

Following the taxonomy of [5], predictive uncertainty is commonly divided into epistemic and aleatoric components. Epistemic uncertainty, or model uncertainty, arises from limited data coverage, insufficient exploration, or model misspecification. In RL, it reflects the agent's lack of knowledge when encountering unfamiliar or out-of-distribution (OoD) states. Evidential Neural Networks (ENNs) provide a computationally efficient way to estimate epistemic uncertainty through a single forward pass [6]. Aleatoric uncertainty, in contrast, captures irreducible variability due to sensor noise, load fluctuations, and RES generation. Common approaches to model aleatoric uncertainty include likelihood-based methods such as NGBoost [7] and quantile regression [8]. Epistemic uncertainty has been incorporated into RL to improve exploration and safety, for example, through bootstrapped Deep Q-Networks (DQNs) that trigger demonstrator queries [9] or ensembles of distributional critics for risk-sensitive control [10]. In power systems, however, uncertainty-aware RL remains relatively unexplored. The L2RPN competitions introduced operator alarms as part of the evaluation framework [11], but many solutions relied on rule-based heuristics or expert knowledge [2]. More recently, supervised learning models have been proposed to predict grid failures at short horizons [12]. Uncertainty estimates are also increasingly used in Safe RL to constrain exploration and reduce the risk of operational-limit violations [13].

This paper proposes a framework for assessing and predicting the reliability of pre-trained RL agents used for power grid congestion management, designed to support human operators by providing early warnings when AI agent decisions may become unreliable. The main contributions of this work are twofold. First, it introduces an AI agent reliability prediction framework that jointly combines epistemic and aleatoric uncertainty estimates with indicators of power system operational complexity to predict the probability of RL agent failure over future time horizons (e.g., the next hour). Second, it investigates the development of inherently interpretable models that help human operators understand the conditions under which AI recommendations are likely to fail. To this end, the work explores two approaches: (i) a novel Large Language Model

(LLM) based approach to distill machine learning models into fully interpretable representations, including natural language explanations, and (ii) the use of decision trees as an alternative to high-performing but less transparent models such as tree ensembles. In contrast to state-of-the-art approaches (e.g., [2], [12]), our method explicitly incorporates both epistemic and aleatoric uncertainty and explores interpretable rule extraction to provide additional insights into the conditions under which AI agent failures occur. Moreover, while this work is centered on RL, the proposed methodology and principles are not limited to this setting and can be transferred to other AI paradigms and deployment configurations.

II. Problem Formulation

A power grid is considered, whose operational state at a discrete time step t is denoted by $s_t \in \mathcal{S}$, where $\mathcal{S}$ represents the state space comprising grid topology, power injections, and line flows. Topology and generator redispatch actions are determined by a pre-trained RL agent governed by a fixed deterministic policy π:

$$\pi : \mathcal{S} \to \mathcal{A}, \qquad a_t = \pi(s_t), \tag{1}$$

where π is treated as a "black box" policy and is not modified or re-trained in this work, and $a_t \in \mathcal{A}$ represents the action taken at time t, given state s_t.

The objective is not to optimize the control performance of the RL agent, but rather to predict the probability of operational failure introduced by deploying this policy. This prediction must account for the stochasticity of the power grid resulting from two distinct sources: epistemic uncertainty (u_t) and aleatoric uncertainty ($\sigma_{a,t}$). For any given state s_t, we aim to predict, at a lead time $t+k$ (where k is the prediction horizon length) and using only the information available at time step t, the probability that the RL agent's recommendation fails to solve a congestion problem at time $t + k$, denoted by $\hat{p}_{fail,t+k|t}$.

Let $Y_{t+k} \in \{0, 1\}$ be a binary variable representing a failed recommendation at time $t + k$, where $Y_{t+k} = 1$ corresponds to a failure event. The objective is to estimate the conditional probability:

$$\hat{p}_{fail,t+k|t} = \mathbb{P}(Y_{t+k} = 1 \mid \mathbf{v}_t), \tag{2}$$

where $\mathbf{v}_t$ represents the vector of features available at time t, formally defined as:

$$\mathbf{v}_t = \left[\mathbf{x}_{topo} \oplus \mathbf{x}_{grid}^{(t)} \oplus \mathbf{x}_{stress}^{(t)} \oplus \mathbf{x}_{grid}^{(t+k)} \oplus \mathbf{x}_{stress}^{(t+k)} \oplus \mathbf{u}_{pred}\right]. \tag{3}$$

The component $\mathbf{x}_{topo} \in \mathbb{Z}$ denotes the integer categorical identifier of the disconnected line l (one integer per line) at time $t + k$, capturing local topological dependencies. The vectors $\mathbf{x}_{grid}^{(t)} \in \mathbb{R}^4$ and $\mathbf{x}_{stress}^{(t)} \in \mathbb{R}^4$ characterize the global operating point and the grid "stress" considering all transmission lines $\mathcal{L}$ at the current time step t, respectively.

Specifically, $\mathbf{x}_{grid}^{(t)}$ comprises total active and reactive load (P_L, Q_L), total generation (P_G), and the load-to-generation ratio $(r_{LG} = P_L/P_G)$, while $\mathbf{x}_{stress}^{(t)}$ includes the maximum line loading $(\max_i \rho_i)$, average loading $(\bar{\rho})$, variance (σ_ρ^2), and the number of critically loaded lines $N_{crit} = |\{i : \rho_i \geq 0.95\}|$.

The vectors $\mathbf{x}_{grid}^{(t+k)} \in \mathbb{R}^3$ and $\mathbf{x}_{stress}^{(t+k)} \in \mathbb{R}^4$ are the predicted counterparts of the above, estimated for the pre-contingency grid state at the target horizon $t + k$, capturing the anticipated operating conditions immediately before a potential $N-1$ line disconnection event. Finally, $\mathbf{u}_{pred} \in \mathbb{R}^5$ concatenates three aleatoric components ($\sigma_{a,t}$ for P_L, Q_L, P_G) and two epistemic uncertainty estimates (at t and $t+k|t$). The ground truth labels Y_{t+k} are obtained via a contingency analysis projected to a one-hour-ahead horizon $(t + 12)$ with realized values.

The central hypothesis of this approach is that incorporating aleatoric and epistemic uncertainty estimation into a supervised classifier enriches the model, thereby enabling more effective discrimination of situations in which the RL agent produces low-confidence recommendations.

III. Uncertainty Modeling

A. Modeling Epistemic Uncertainty for RL Agents

Epistemic uncertainty, u, reflects the RL agent's confidence in a given grid state by measuring its similarity to training conditions. It remains low under familiar operating regimes but increases for OoD states, such as post-contingency topologies, indicating that the agent's recommended actions may be unreliable. For uncertainty quantification, the ENN model [6] is employed, leveraging Dempster–Shafer theory and subjective logic. Unlike classical neural networks, which produce a single probability estimate per class, ENN treats predictions as higher-order probability distributions. Specifically, the model is trained to parameterize a Dirichlet distribution. The Dirichlet distribution over the K possible action classes (i.e., the size of the curated topology action set), whose concentration parameters $\alpha = [\alpha_1, \ldots, \alpha_K]$ are obtained from an evidence vector $e = [e_1, \ldots, e_K]$ such that $\alpha_k = e_k + 1$.

The total accumulated evidence $S = \sum_{k=1}^{K} \alpha_k$ reflects the density of training states like the current state: the larger S is, the more the model recognizes the state as familiar. Epistemic uncertainty is then defined as [6]:

$$u = K/S \tag{4}$$

producing an interpretable scalar in the interval $(0, 1)$. In familiar states, the evidence is high $S \gg K$ and $u \to 0$; in unknown states, the evidence is minimal $(e_k \approx 0$, so $S \approx K)$ and $u \to 1$.

This formulation has a practical advantage: by directly parameterizing the predictive distribution, ENN avoids the need for sampling methods such as Monte Carlo Dropout or ensemble models, resulting in significantly faster inference that is more suitable for real-time applications in power systems.

The ENN is trained via behavior cloning on the state-action pairs visited by the RL agent during training, mapping each state s_t to the agent's selected topology action. Since it only ever observes the regions of the state space the agent has actually experienced, its epistemic uncertainty acts as a measure of state familiarity: low when s_t resembles training

conditions, and high when the agent operates outside its experience. It does not replace the policy: the agent issues the control action, while the ENN runs in parallel to flag the grid states in which that action is least reliable.

B. Aleatoric Uncertainty in Load and Generation Forecasts

To estimate aleatoric uncertainty, which captures the inherent stochasticity of the data, a two-stage regression approach is employed to model heteroscedastic variance. First, a mean prediction model based on Histogram-based Gradient Boosting (HGB) [14] is trained. The squared residuals of the training set predictions are then computed to approximate the local variance. A second HGB model is subsequently trained to predict these residuals using a Poisson loss function, which was chosen because the squared residual is non-negative and right-skewed, yielding stable training and positive variance estimates without additional assumptions. More advanced probabilistic methods, such as heteroscedastic likelihoods or NGBoost [7], are left for future work. This choice is consistent with probabilistic boosting approaches that learn variance directly to improve calibration [7].

IV. RL Agent Failure Prediction and Interpretability

A. RL Agent Failure Prediction System

The complete system[1] for predicting, e.g., one hour-ahead ($t + 12$), situations in which the RL agent is unable to provide a recommendation that resolves a congestion problem is depicted in Fig. 1. The first stage consists of a multi-output HGB regression model that predicts future grid states, namely active (P_L) and reactive power loads (Q_L), active power generation (P_G) at a one-hour-ahead horizon ($t + 12$). The input features are historical observations at four time lags - current step (t), one-hour ($t-12$), one-day ($t-288$), and one-week ($t-2016$), assuming a 5-minute resolution - along with calendar variables (hour, day of the week) encoded via a Fourier series. The resulting forecasts of power injections are then used to construct the future grid operating state s_{t+12} through a power flow solver (*LightSimBackend* [15]). Ground-truth labels are obtained via $N-1$ contingency analysis on the observed grid state, where the RL agent's ability to resolve the resulting congestion determines success or failure.

Following the contingency analysis, a HGB classifier is trained to distinguish between states in which the RL agent provides recommendations that successfully resolve congestion problems and those in which it fails to do so. This classifier acts as an early warning system, triggering alerts when a potential failure is anticipated. The HGB model leverages the following inputs to generate a binary risk alert:

- *Power balance:* total active and reactive load, total generation, and the load-to-generation ratio, incorporating both the measurements at the current time step (t) and the forecasts for the one-hour-ahead horizon ($t+12$, prior to disconnection).

[1]Code available at: https://github.com/ainetus/RL_agent_failure_forecast

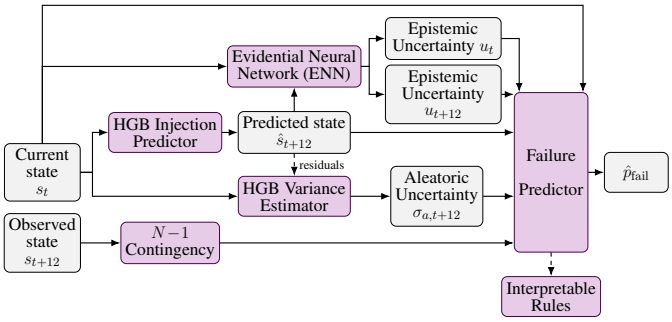

Fig. 1: Workflow of the proposed failure prediction system.

- *Grid "stress":* statistics of line loading rates ρ, including maximum, average, variance, and the number of critical lines ($\rho \geq 0.95$). These statistics are extracted for both the current state (t) and the predicted future state ($t+12$).
- *Uncertainty:* predicted aleatoric uncertainty derived from the HGB variance estimator (trained on the same historical observation features to capture the inherent stochasticity of the prediction model), and epistemic uncertainty estimated by the ENN. Both uncertainty metrics are evaluated for the current (s_t) and future (s_{t+12}) grid states.
- *Grid contingencies:* identifier of the disconnected line.

To assess performance in this class-imbalanced setting, the evaluation emphasizes not only overall accuracy but also the trade-off between overlooked failures and false alarms, using two complementary metrics: the *False Alarm rate (FA)* and the *Oversight Rate (OVR)*. These metrics are derived from the standard confusion matrix components—True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)—and defined as: $FA = FP/(TN + FP)$; $OVR = FN/(TP + FN)$.

B. Generation of Interpretable Rules

The final stage of the framework (Fig. 1) builds an interpretable surrogate for the HGB failure predictor from Section IV-A. Although the HGB gives accurate one-hour-ahead predictions of RL agent failures, its decision process is not readily interpretable. Two approaches are explored: a decision tree (DT) trained on the same features and data, producing inherently interpretable rules that may nonetheless grow large; and an LLM-based extraction (Algorithm 1) that distills the "black-box" HGB into a compact set of human-readable `if-then` rules. Both are trained independently per disconnected line, enabling line-specific explanations.

A balanced subset $Data_{bal}$ is first constructed from the training data, sampling a 2:1 ratio between normal-operation and failure cases to reduce the majority-class bias. Class-separated medians and quartiles, denoted $Stats$, are then computed to provide numerical anchors for threshold selection, since LLMs require explicit numerical guidance to identify meaningful decision boundaries. For each line, the input features are ranked by applying SHAP [16] on the HGB model (SHAP-Rank in Algorithm 1), retaining only

the top k ($k \in \{3, \ldots, 8\}$, selected per line). Initial rules are generated from $Stats$ as single-feature thresholds and AND/OR combinations of these top features, and the best-performing one is retained as R_{best} with score S_{best}.

The refinement proceeds through a dual Generator-Critic loop. A history-based $DetectOscillation$ check identifies whether candidates alternate between overly conservative behavior (excessive false alarms) and overly permissive behavior (missed failures); when detected, a flag Osc restricts the *Generator* LLM ($\mathrm{Gen_{LLM}}$) to local single-threshold adjustments. The *Generator* proposes a candidate rule R_{cand} as executable Python *if-else* logic. The rules combine AND and OR conditions across two distinct failure paths: one based on the current grid-state and uncertainty features, capturing out-of-distribution situations, and one based on forecasted features, capturing anticipated failures. The prompt[2] includes $Data_{bal}$, $Stats$, the selected features, the rule history, and misclassified instances WC, where false negatives are prioritized as missed failure warnings.

Each candidate undergoes syntactic and structural validation (*Validate*). If violations are found, a *Repair* LLM ($\mathrm{Repair_{LLM}}$) corrects the rule using its error trace, inspired by LLM self-debugging [17]. Once validated, the rule is scored on the training set using Accuracy, F_2-score, OVR, and FA; the retained rule is evaluated on the held-out test set for comparability with the DT and HGB. The *Critic* LLM ($\mathrm{Critic_{LLM}}$) then provides targeted feedback for the next iteration. The Generator, Critic, and Repair LLMs are instances of the same base model queried with role-specific prompts.

Rules are ranked by the F_2-score, which prioritizes recall, while a guard discards degenerate rules that classify nearly all inputs as positive ($FA \geq 0.90$). If a candidate improves S_{best}, it replaces R_{best}; otherwise $StagCount$ increments. The output is the best rule across all iterations, with its natural-language translation describing when the RL agent is predicted to fail.

Algorithm 1 Dual LLM-based RL failure prediction

Require: Dataset $\mathcal{D}$, HGB model $\mathcal{HGB}$, Iterations (N)
Ensure: Best interpretable rule R_{best}
$Train, Test \leftarrow$ SequentialSplit($\mathcal{HGB}.Predict(\mathcal{D})$)
$Data_{bal}, Stats \leftarrow$ SampleBalancedData($Train$)
$Feats \leftarrow$ SHAP-Rank($\mathcal{HGB}, Train$), $k \in \{3, \ldots, 8\}$
$R_{best}, S_{best} \leftarrow$ Evaluate(InitRules$_{\text{AND/OR}}(Stats, Feats), Train$)
$History \leftarrow \emptyset$, $StagCount \leftarrow 0$
for $iter = 1$ **to** N **do**
  $Osc \leftarrow$ DetectOscillation($History$, window $= 4$)
  $WC \leftarrow MisclassifiedCases(R_{best}, Train)$
  $R_{cand} \leftarrow \mathrm{Gen_{LLM}}(Data_{bal}, Stats, Feats, WC, History, Osc)$
  if not Validate(R_{cand}) **then**
    $R_{cand} \leftarrow \mathrm{Repair_{LLM}}(R_{cand})$
  end if
  $Metrics_i \leftarrow$ ComputeMetrics($R_{cand}(Train.X), Train.Y$)
  $S_i \leftarrow F_2$-score($Metrics_i$)
  $Feedback \leftarrow \mathrm{Critic_{LLM}}(R_{cand}, Metrics_i, Stats, Osc)$
  $History$.Append($R_{cand}, Metrics_i, S_i, Feedback$)
  if $S_i > S_{best}$ **then**
    $R_{best}, S_{best} \leftarrow R_{cand}, S_i$
    $StagCount \leftarrow 0$
  else
    $StagCount \leftarrow StagCount + 1$
  end if
end for
return R_{best}, evaluated on $Test$

V. Numerical Results

A. Test System

The proposed approach was evaluated using the Grid2Op digital environment [18], which provides a standardized environment for training and evaluating AI-based congestion management agents. The experiments were conducted in the `l2rpn_icaps_2021_small` environment, which is based on the IEEE 118-bus system and includes 36 substations, 59 power lines, 22 generators, and 37 loads, some of which represent interconnections with an external grid.

The RL agent evaluated in this study is the `CurriculumAgent` [19], which follows a Teacher–Tutor–Junior–Senior learning framework. First, a Teacher agent performs an exhaustive search over topology actions to identify a reduced set of effective actions. Using this action set, a rule-based Tutor agent generates observation–action trajectories, which are then used to train a Junior neural network through imitation learning. Finally, a Senior RL agent is initialized with the Junior network weights and further optimized using proximal policy optimization.

As discussed in Section IV-A, the method relies on forecasted grid conditions, whose error increases gradually and approximately linearly from 5 to 60 minutes due to recursive predictions, indicating stable multi-step forecasting over the considered horizon. For active power generation (P_G), the mean RMSE increases from 1.51 MW at 5 minutes to 7.18 at 60 minutes. Load errors remain lower, with P_L rising from 0.87 to 4.51 and Q_L from 0.22 to 0.77, reflecting the higher variability of RES-driven generation. The 95th-percentile errors reveal higher variance at some buses over longer horizons, but growth remains bounded (18.95 for P_G and 21.28 for P_L at 60 minutes) without instability.

B. Quality of Epistemic Uncertainty Estimation

Epistemic uncertainty is assessed from two complementary perspectives: calibration quality and its effectiveness as an early-warning signal for RL agent failures within a 60-minute horizon.

The calibration of the ENN confidence estimates is first evaluated before using them as a warning signal. Calibration is measured using the Expected Calibration Error (ECE), which quantifies the average gap between predicted confidence and empirical accuracy. The ground-truth labels correspond to the topological actions selected by the `CurriculumAgent` Tutor, which the ENN is trained to imitate through behavior cloning. The ENN achieves an ECE of 0.0297; since a perfectly calibrated model has ECE=0, this indicates that its confidence estimates deviate from empirical accuracy by only

[2]Prompts available at: https://github.com/ainetus/RL_agent_failure_forecast

TABLE I: AUROC of the ENN's epistemic uncertainty under different adversarial operating scenarios.

Scenario	Environment Condition	Mean U	AUROC
Standard	Nominal operation (no targeted attacks)	0.4594	0.6315
HeavyAttack2	Frequent disconnection of critical lines	0.4635	0.6564
HeavyAttack1	Severe but less frequent disconnections	0.4698	0.6686

2.97% on average, showing limited overconfidence despite the extreme multi-class setting.

The practical usefulness of epistemic uncertainty was then evaluated as a warning signal for failure in the 60-minute horizon ($t + 12$). This is a highly imbalanced prediction problem, since these failures represent only 11.1% of the 261,272 analyzed samples. For this reason, the Area Under the Precision-Recall curve (AUPR) is particularly informative, as it measures whether the uncertainty score provides meaningful discrimination beyond the low base rate of failures. The ENN's epistemic uncertainty achieved an AUPR of 0.1521, which, given the highly imbalanced nature of the problem (failures account for only 11.1% of the samples), is above the random baseline of 0.1114.

To further analyze its behavior, the predictive performance of ENN was evaluated under different adversarial attack scenarios applied to the grid using the Area Under the Receiver Operating Characteristic curve (AUROC). The results in Table I corroborate the OoD detection capability of the ENN. During standard operation, corresponding to nominal conditions without adversarial disconnections of critical power lines, the discriminative power is moderate, with an AUROC of 0.6315. This is expected, since under routine operating conditions, the RL agent is more likely to encounter states similar to those observed during training.

The warning signal becomes more informative under *HeavyAttack* scenarios, where critical line disconnections drive the system further from the training distribution. These unfamiliar grid states increase epistemic uncertainty, providing a stronger indication of potential failures. Under *HeavyAttack2*, which involves more frequent attacks on critical lines, the AUROC increases to 0.6564. Under *HeavyAttack1*, characterized by more severe attacks that push the grid closer to critical limits, the AUROC further increases to 0.6686.

Furthermore, as shown in Table I, the mean epistemic uncertainty (U) scales consistently with the severity of the adversarial perturbations. The baseline uncertainty is lowest during standard operation ($U = 0.4594$) and reaches its highest value under the most severe *HeavyAttack1* scenario ($U = 0.4698$). Since u is bounded by the action-set size K, its absolute value remains modest in this large multi-class setting. The relevant signal is the increase under OoD conditions ($0.4594 \rightarrow 0.4698$) and the corresponding AUROC. This provides numerical evidence that the ENN correctly perceives these aggressively attacked topologies as OoD states. The combination of the highest mean uncertainty and the highest discriminative power (AUROC of 0.6686) under *HeavyAttack1* suggests that the epistemic uncertainty can provide an informative early-warning signal.

TABLE II: Performance of the HGB models for one-hour-ahead RL agent failure prediction (mean $\pm$ std).

Configuration	F_2-score	Accuracy	OVR	FA
Baseline (grid variables)	70.8 ± 0.9	85.7 ± 0.3	7.7 ± 0.9	14.9 ± 0.3
Epistemic-only	35.4 ± 1.1	42.7 ± 0.6	20.1 ± 1.4	60.8 ± 0.6
Grid + aleatoric	73.6 ± 0.9	87.6 ± 0.4	7.1 ± 0.7	12.9 ± 0.4
Grid + aleat. + epistemic	76.5 ± 0.6	89.5 ± 0.3	6.8 ± 0.8	10.8 ± 0.3

C. Failure Prediction for the RL Agent

This subsection evaluates one-hour-ahead prediction of the RL agent failure under $N - 1$ contingencies. This evaluation must capture not only overall accuracy, but also the ability to detect failures while limiting unnecessary alarms. For this reason, performance is assessed using Accuracy, F_2-score, OVR, and FA. To quantify the contribution of uncertainty-aware features, four HGB classifier configurations were compared: (1) *baseline* model (grid variables), (2) *epistemic-only* (epistemic uncertainty), (3) *without epistemic uncertainty* (grid variables + aleatoric uncertainty), and (4) *complete* model (grid variables + aleatoric + epistemic). Table II summarizes the model performance in predicting $\hat{p}_{fail,t+12|t}$, reporting the mean and standard deviation over 10 repeated stratified hold-out splits.

The *baseline* model provides a reasonable trade-off between failure detection and alarm control, but exhibits a relatively high FA. Adding aleatoric uncertainty improves Accuracy and F_2-score, while reducing both the FA and OVR. In contrast, the *epistemic-only* model shows severe performance degradation, achieving the lowest Accuracy and F_2-score, along with a significantly increased FA. This suggests that epistemic uncertainty alone does not provide sufficient information to discriminate between failure and non-failure states, and is poorly calibrated as a standalone predictor. However, when combined with grid-related and aleatoric uncertainty features, epistemic uncertainty significantly improves predictive performance. The *complete* model achieves the best overall trade-off, attaining the highest F_2-score and Accuracy, while simultaneously reducing both OVR and FA. These results show that epistemic uncertainty becomes highly informative when combined with grid-level variables.

D. Interpretable Failure Prediction

The per-line interpretable rule extraction of Section IV-B was applied to the 10 contingencies of the `l2rpn_icaps_2021_small` environment. The LLM experiments were performed using the `gemma4-31b` LLM model. Table III compares a DT, the LLM-based extraction, and the HGB. DT and HGB are reported as mean $\pm$ std over 10 repeated stratified hold-out splits; the LLM is reported as the mean $\pm$ std over 10 random seeds at $\tau = 0.7$, where τ controls the randomness of candidate rule generation.

TABLE III: Interpretable models against the HGB reference.

Model	F_2-score	Accuracy	OVR	FA
LLM rule†	**73.8±0.3**	**89.4±0.1**	**13.8±0.8**	**11.3±0.3**
DT	72.4±0.7	87.1±0.5	14.4±0.8	12.8±0.5
HGB	76.5±0.6	89.5±0.3	6.8±0.8	10.8±0.3

†mean $\pm$ std over 10 random seeds at $\tau = 0.7$.

The LLM-derived rules match or outperform the DT across all reported metrics while remaining substantially more compact. They achieve a higher F_2-score (73.8% vs. 72.4%) and accuracy (89.4% vs. 87.1%), together with lower OVR (13.8% vs. 14.4%) and FA (10.3% vs. 12.8%) rates, approaching the HGB in terms of F_2-score and accuracy. This performance is obtained using only a few `if-then` conditions per contingency, whereas the DT comprises 737 leaves with a maximum depth of 14, and the HGB is an opaque ensemble of approximately 592 trees, making both models difficult to inspect or validate comprehensively. Although the HGB achieves the lowest OVR (6.8%), indicating superior sensitivity to the rarest failures, the LLM-derived rules capture the dominant decision factors through per-line SHAP-based feature selection and disjunctive (OR) conditions. Consequently, while they do not reproduce the fine-grained decision boundary of the 21-feature HGB, they provide concise, operator-readable explanations that summarize the key factors leading to RL agent failures rather than attempting to replace the HGB classifier.

The LLM offers a compact `if-else` symbolic model for each contingency that an operator can easily inspect, audit, and translate into natural language. An illustrative rule for contingency in line `34_35_110` is: *the RL agent is predicted to fail when the load-to-generation ratio at t is ≤ 0.986 and either the mean aleatoric reactive power load uncertainty for $t+12|t$ is ≥ 0.185 or the AI agent epistemic uncertainty for $t+12|t$ is ≥ 0.803.* Here the low load-to-generation ratio defines the operating regime, within which failure is flagged by either elevated aleatoric uncertainty in the reactive power load or high forecasted epistemic uncertainty, the latter signaling anticipated OoD conditions.

In mission-critical tasks such as congestion management, this trade-off between interpretability and accuracy should be carefully evaluated. While OVR should be minimized to avoid missed failures, the primary role of interpretability is to provide additional insights to the human operator rather than to replace a high-performing predictive model such as the HGB classifier. A practical deployment could therefore combine two complementary models: a highly accurate HGB model for failure prediction and an interpretable surrogate that provides concise explanations, helping operators understand failure modes and the key factors that affect the AI agent's decisions.

VI. Conclusions

AI-based solutions for critical infrastructures such as power grids must operate under uncertainty while combining domain knowledge with data-driven learning. However, most AI agent frameworks lack mechanisms to recognize when their recommendations may become unreliable. This paper addresses this limitation by incorporating epistemic uncertainty estimation into RL agents, enabling uncertainty-aware decision-making and confidence assessment. Such capabilities are essential in environments subject to incomplete knowledge, evolving operating conditions, and distribution shifts. Results show that epistemic uncertainty is a useful indicator of agent failure risk, allowing models trained on historical data to anticipate failures one hour ahead with an F_2-score of 76.5% and an OVR below 7%, supporting timely deferral to human operators. In addition, interpretable surrogate models, notably the proposed LLM-based rule extraction, complement the failure predictor with compact, human-readable explanations whose predictive performance matches or exceeds a decision tree baseline while remaining substantially easier to understand and inspect, facilitating operator trust and adoption. Future work will investigate conformal prediction for uncertainty quantification and explore the concept of evolving symbolic model [20] to build accurate interpretable models.

References

[1] M. Mussi, A. M. Metelli, M. Restelli, G. Losapio, R. J. Bessa, D. Boos, *et al.*, "Human-AI interaction in safety-critical network infrastructures," *iScience*, vol. 28, p. 113400, Sept. 2025.

[2] A. Pavão, A. Marot, *et al.*, "AI challenge for safe and low carbon power grid operation," *Energy and AI*, vol. 22, p. 100564, Dec. 2025.

[3] E. van der Sar, A. Zocca, and S. Bhulai, "Optimizing power grid topologies with reinforcement learning: A survey of methods and challenges," *Found. Trends Electr. Energy Syst.*, vol. 9, pp. 1–119, Aug. 2025.

[4] M. R. Endsley, "Ironies of artificial intelligence," *Ergonomics*, vol. 66, pp. 1656–1668, Nov. 2023.

[5] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Mach. Learn.*, vol. 110, pp. 457–506, 2021.

[6] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *NeurIPS*, vol. 31, 2018.

[7] T. Duan, A. Avati, D. Y. Ding, S. Khan, A. Y. Ng, and P. Basu, "Ngboost: Natural gradient boosting for probabilistic prediction," in *Int. Conf. Mach. Learn. (ICML)*, pp. 2690–2700, PMLR, 2020.

[8] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecast.*, vol. 32, no. 3, pp. 914–938, 2016.

[9] F. L. D. Silva, P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "Uncertainty-aware action advising for deep reinforcement learning agents," in *AAAI Conf. Artif. Intell. (AAAI-20)*, 2020.

[10] M.-H. Ibrahim, S. Lecoeuche, J. Boonaert, and M. Batton-Hubert, "Uncertainty quantification for efficient and risk-sensitive reinforcement learning," in *IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2023.

[11] A. Marot *et al.*, "Learning to run a power network with trust," *Electr. Power Syst. Res.*, vol. 212, p. 108487, 2022.

[12] M. Lehna, M. Hassouna, D. Degtyar, S. Tomforde, and C. Scholz, "Fault detection for agents on power grid topology optimization: A comprehensive analysis," in *Eur. Conf. Mach. Learn. Knowl. Discov. Databases (ECML PKDD), Workshop Mach. Learn. Sustain. Power Syst. (ML4SPS)*, Sept. 2024.

[13] T. Su, T. Wu, J. Zhao, A. Scaglione, and L. Xie, "A review of safe reinforcement learning methods for modern power systems," *Proc. IEEE*, vol. 113, pp. 213–255, Mar. 2025.

[14] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html, 2024. Accessed: Dec. 31, 2025.

[15] B. Donnot, "Lightsim2grid - A C++ backend targeting the Grid2Op platform." https://GitHub.com/Grid2Op/lightsim2grid, 2020.

[16] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[17] C. S. Xia and L. Zhang, "Automated program repair in the era of large pre-trained language models," in *45th Int. Conf. Softw. Eng. (ICSE 2023)*, pp. 1482–1494, 2023.

[18] J. Paulos, P. R. Silva, R. J. Bessa, A. Marot, J. Dejaegher, and B. Donnot, "Generation of power network operating scenarios for an AI-friendly digital environment," in *IEEE PowerTech Conf.*, (Kiel, Germany), 2025.

[19] M. Lehna, J. Viebahn, A. Marot, S. Tomforde, and C. Scholz, "Managing power grids through topology actions: A comparative study between advanced rule-based and reinforcement learning agents," *Energy and AI*, vol. 14, p. 100276, 2023.

[20] F. S. Fernandes, R. J. Bessa, and J. P. Lopes, "Evolving symbolic model for dynamic security assessment in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 13, no. 4, pp. 1113–1126, 2025.