

Risk-Guided Hybrid Falsification for Safety Verification of AI-Enabled Cyber-Physical Systems

Varad Deepak Shahane¹, Prof M. N. Pimple², Prof A. R. Kardile³, Dr. H. B. Jadhav⁴

Student, Computer Department¹

Assistant Professor, Computer Department^{2,3}

Head of Department, Computer Department⁴

Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, India

Abstract: *AI-enabled cyber-physical systems (CPS) combine software-based decisions with physical processes. Their complex and sometimes unpredictable behaviour make it difficult to discover inputs that violate safety requirements. Conventional temporal-logic falsification tools often use robustness-guided optimization, but may spend many tests in similar regions and can struggle with high-dimensional input spaces. This paper presents a practical risk-guided hybrid falsification framework that combines robustness scores with a simple reinforcement-learning feedback loop. The system first runs baseline scenarios, measures how close each run comes to violating a temporal property, and then prioritizes input regions that appear risky or insufficiently explored. A guided test generator proposes further scenarios, while a validation stage replays candidate counterexamples before they are reported. The framework builds on established falsification concepts and introduces risk prioritization as an organizing layer rather than claiming a completed implementation. A workflow, system architecture, and comparison with a conventional approach are presented. Evaluation is proposed using both traditional and AI-enabled CPS benchmarks, with detection rate, time to first violation, test diversity, and simulation cost as measures.*

Keywords: Cyber-physical systems, temporal logic, falsification, robustness, reinforcement learning, safety testing

I. INTRODUCTION

Cyber-physical systems connect computation, communication, and physical processes. Examples include autonomous vehicles, industrial control systems, and medical devices. The addition of artificial intelligence can make these systems more adaptable, but it also introduces behaviours that are harder to predict using conventional models alone. Safety verification must therefore consider both the physical dynamics and the decisions made by learned components. Temporal-logic falsification is a testing approach that searches for an input sequence under which a required property becomes false. Instead of proving that a system is safe for every possible input, falsification tries to find a concrete counterexample that can be inspected and reproduced. Tools such as S-TaLiRo and Breach use robustness measures to guide the search toward likely violations.

However, AI-enabled models may have many input variables, interconnected components, and stochastic behaviours. A search algorithm can repeatedly explore familiar regions or miss rare but important scenarios. The base study by Yusuf discusses these challenges and proposes combining stochastic optimization with reinforcement learning to improve the discovery of temporal-property violations. This paper develops that general direction into a straightforward risk-guided workflow. Its main idea is to use the observed robustness score, repeated failure patterns, and exploration history to



decide which input regions should receive additional tests. The proposal is conceptual and requires experimental evaluation before any performance improvement can be claimed.

II. PROBLEM STATEMENT AND OBJECTIVES

The main problem is the inefficient discovery of safety-property violations in AI-enabled CPS when the test space is large and system behaviour is difficult to predict. A conventional optimizer may identify useful cases, but its search can be costly when a violation occurs only in a narrow region of the input space.

The objectives of the proposed framework are:

- (1) represent system requirements using Metric or Signal Temporal Logic where suitable;
- (2) run baseline scenarios within defined input limits;
- (3) calculate a robustness score for each simulation;
- (4) prioritize scenarios that are close to violation or explore under-tested regions; and
- (5) validate and document any candidate counterexample through replay.

III. EXISTING SYSTEM

In a conventional robustness-guided falsification system, the user defines a model, a set of input constraints, and a temporal-logic property. A scenario generator supplies inputs to the model. The monitor evaluates the resulting signal trace and returns a robustness value. An optimizer uses this value to propose the next test, with the goal of finding a trace that violates the property.

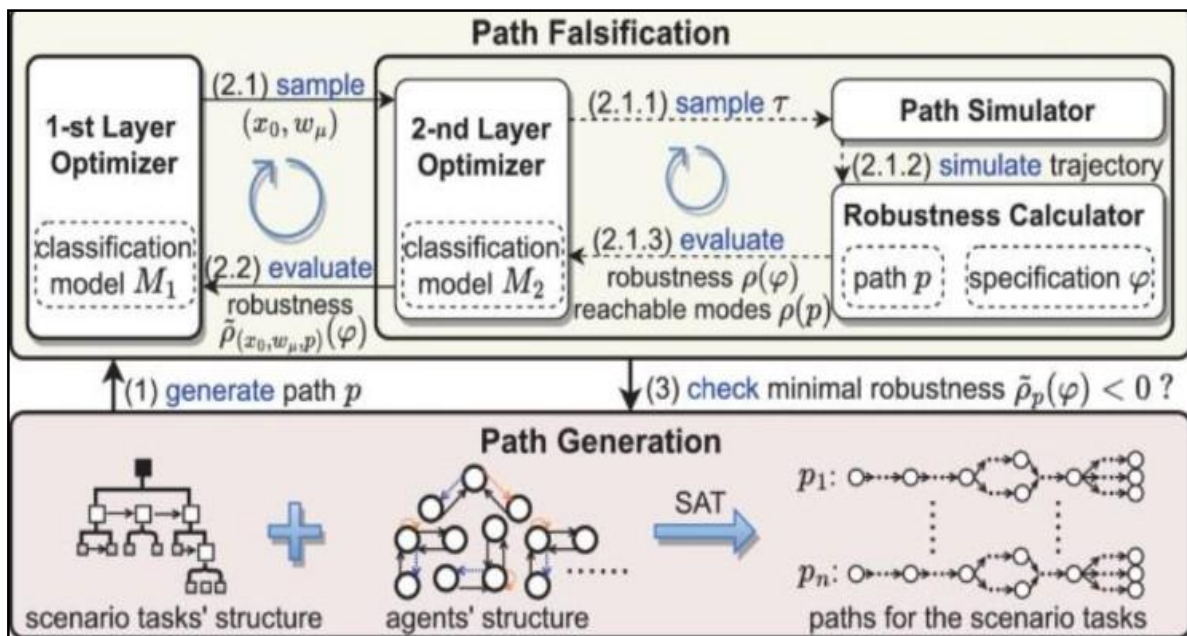


Fig. 1. General flow of an existing robustness-guided falsification system.

The main advantage of this approach is that testing is guided by a measurable distance from satisfaction or violation rather than relying only on random sampling. Its limitations become more visible when the system is high-dimensional or AI-driven. Search quality may depend on the selected optimizer, model behaviour, and available computational budget. The base paper reports preliminary observations that AI-driven models can require longer falsification times and show variable success rates compared with traditional models [7]. These observations motivate, but do not by themselves prove, the benefit of the proposed design.



IV. PROPOSED SYSTEM

The proposed system adds a risk-prioritization layer between robustness evaluation and the next-test decision. It does not replace the simulator or temporal monitor. Instead, it organizes the search using three practical signals:

- robustness magnitude, indicating how close a run is to a property boundary;
- recurrence of similar near-failure patterns; and
- exploration history, indicating whether a region has been tested often or rarely.

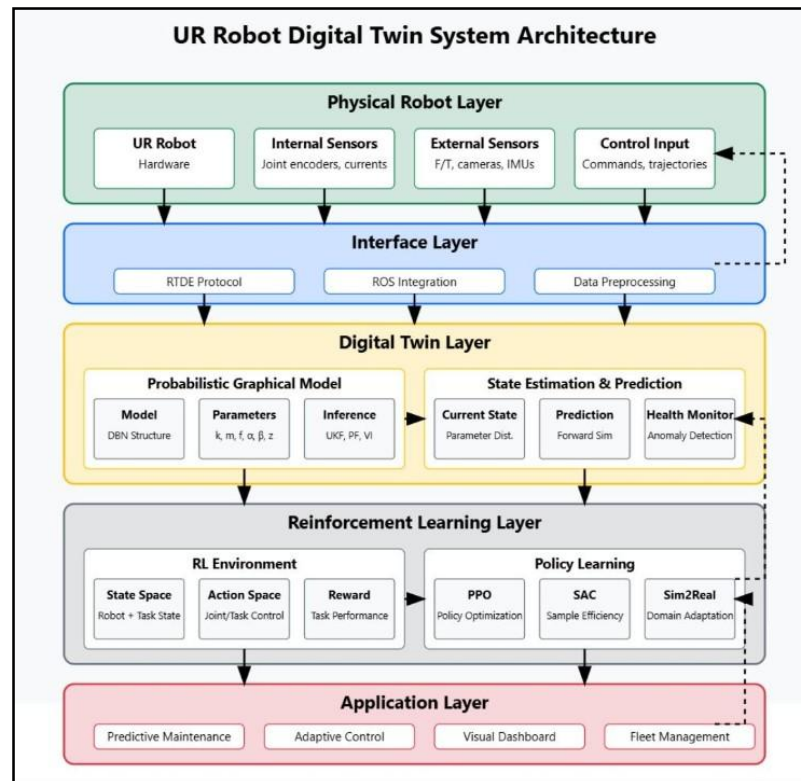


Fig. 2. Proposed architecture for risk-guided hybrid falsification.

A. Main Components

- 1) Scenario generator: Creates bounded input sequences such as speed, sensor values, environmental conditions, or control requests. The exact variables depend on the selected CPS model.
- 2) CPS simulator: Executes the scenario against the physical model and any AI decision component. Inputs and outputs are recorded so that a suspicious result can be replayed.
- 3) Robustness evaluator: Checks the output trace against the temporal property and returns a robustness score. A score near the violation boundary is treated as a signal for further investigation; the sign and interpretation depend on the property and monitor used.
- 4) Risk-based decision layer: Ranks candidate regions using robustness, prior outcomes, and coverage information. This layer helps avoid spending the entire budget on one repeatedly sampled region.
- 5) Guided search and feedback: A stochastic optimizer proposes candidates, while a reinforcement-learning agent can learn which actions tend to lead toward low-robustness or under-explored areas. The agent receives feedback from the simulation and robustness evaluator. If reinforcement learning is not available, the framework can operate with the risk-ranked optimizer alone.



6) Counterexample review: Candidate violations are replayed with the same settings. The result is reported as a counterexample only when the violation is reproducible and the input constraints have been respected.

B. Working Principle

At the beginning, the user specifies the safety property and allowable input ranges. The system runs a small baseline set and stores the scenario, trace, robustness score, and coverage information. After each batch, the risk layer identifies promising regions. The search then generates new scenarios around those regions while preserving some exploration of less-tested inputs. This balance is intended to reduce premature concentration on one search region. Final candidates are independently replayed and summarized for review.

V. WORKFLOW OF THE PROPOSED METHOD

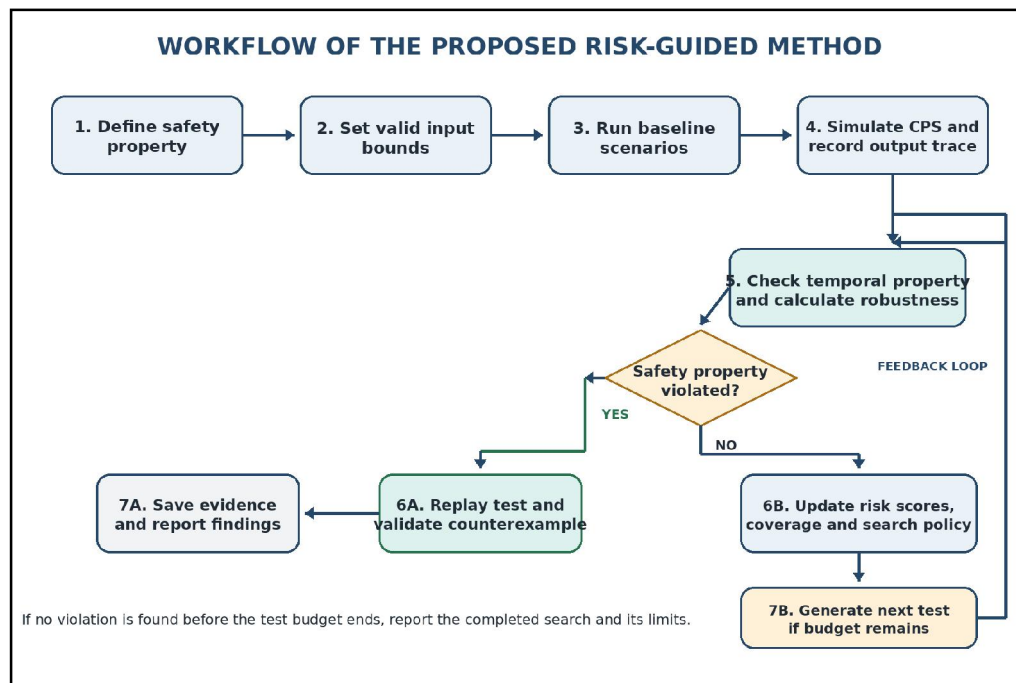


Fig. 3. Step-by-step workflow for the proposed method.

- Step 1 — Define the safety property: Express the required behaviour as a temporal-logic formula, such as maintaining a safe distance or keeping a measured value within a permitted range.
- Step 2 — Set input bounds: Specify valid ranges and constraints for scenario variables so that generated tests remain meaningful.
- Step 3 — Run baseline tests: Execute initial scenarios to understand ordinary system behaviour and establish initial robustness values.
- Step 4 — Measure robustness: Evaluate each output trace against the chosen temporal property and record the score.
- Step 5 — Prioritize risky regions: Select scenarios near the property boundary, recurring near-failures, and under-explored regions for further testing.
- Step 6 — Generate new tests: Use stochastic optimization and, where implemented, reinforcement-learning feedback to propose the next input sequences.
- Step 7 — Validate counterexamples: Replay candidate violations and check whether they persist under the same model configuration and valid constraints.



Step 8 — Report findings: Save the scenario, property, trace, score, model configuration, and outcome so the test can be reproduced.

VI. EVALUATION PLAN

The framework should be evaluated on a collection of traditional and AI-enabled CPS models. Where possible, the same model, property, input bounds, and computational budget should be used for both the conventional baseline and the proposed approach. The benchmark set should include different model structures and levels of input complexity, following the need for broader CPS benchmarks identified in the base study.

The following measures are recommended:

- (1) violation detection rate, defined as the proportion of runs in which a valid counterexample is found;
- (2) time to first valid violation;
- (3) number of simulations used before detection;
- (4) diversity of tested input regions; and

(5) reproducibility of reported counterexamples. Results should be reported separately for traditional and AI-enabled models and across repeated runs, because stochastic methods can vary between executions.

A fair experiment should include a conventional robustness-guided optimizer as the baseline and, if resources permit, an ablation that disables the risk-prioritization layer or reinforcement-learning feedback. No numerical results are claimed in this paper because the proposed framework has not been experimentally implemented or measured here

VII. EXPECTED BENEFITS AND LIMITATIONS

A. Expected Benefits

- **Efficient testing:** Robustness-guided testing helps identify scenarios that are close to violating safety requirements, allowing the available test budget to be used more effectively.
- **Improved coverage:** Coverage history encourages the system to explore less-visited regions instead of repeatedly testing familiar scenarios.
- **Adaptive learning:** The feedback loop allows the framework to learn from previous test results and use that information to guide subsequent test generation.
- **Early fault detection:** Focusing on high-risk scenarios may help identify potential safety violations earlier during the verification process.
- **Systematic verification:** Combining robustness scores, coverage information, and reinforcement learning provides a structured approach to exploring complex AI-enabled cyber-physical systems.

B. Limitations

- **Simulator accuracy:** The framework depends on the accuracy of the simulator. An inaccurate model may produce misleading counterexamples that do not represent actual system behavior.
- **Computational cost:** Reinforcement learning requires additional training, computational resources, and parameter tuning, which may increase the overall verification time.
- **Limited generalization:** A reinforcement learning policy trained on one system may not perform effectively when applied to a different model or operating environment.
- **Risk prioritization:** Risk scores may over-prioritize familiar patterns or previously identified failure regions if the framework does not preserve sufficient exploration of new scenarios.
- **Experimental validation:** The expected benefits are design expectations rather than confirmed empirical findings. Comparative experiments and repeatable reporting are necessary to evaluate the framework's actual effectiveness.



VIII. CONCLUSION

This paper presented a risk-guided hybrid approach for falsifying temporal-logic safety properties in AI-enabled cyber-physical systems. The design combines scenario generation, simulation, robustness evaluation, risk-based prioritization, guided search, and counterexample replay. Compared with a basic robustness-guided flow, the added risk layer explicitly considers previous test outcomes and exploration history. The proposed method offers a practical structure for further implementation, but its effectiveness must be established through benchmark-based evaluation. Future work should implement the workflow, compare it with existing falsification tools, and publish reproducible scenarios and measurements.

REFERENCES

- [1]. J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating physics-based modeling with machine learning: A survey," arXiv preprint arXiv:2003.04919, 2020..
- [2]. R. Rai and C. K. Sahu, "Driven by data or derived through physics? A review of hybrid physics-guided machine learning techniques with cyber-physical system focus," IEEE Access, vol. 8, pp. 71050–71073, 2020.
- [3]. J. Song et al., "When cyber-physical systems meet AI: A benchmark, an evaluation, and a way forward," in Proc. 44th Int. Conf. Software Engineering: Software Engineering in Practice, 2022, pp. 343–352.
- [4]. J. X. Liu, H. Xu, W. Liao, and W. Yu, "Reinforcement learning for cyber-physical systems," in Proc. IEEE Int. Conf. Industrial Internet, 2019, pp. 318–327.
- [5]. Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, "S-TaLiRo: A tool for temporal logic falsification for hybrid systems," in Tools and Algorithms for the Construction and Analysis of Systems, 2011, pp. 254–257.
- [6]. A. Donzé, "Breach, a toolbox for verification and parameter synthesis of hybrid systems," in Computer Aided Verification. Springer, 2010, pp. 167–170.
- [7]. H. U. Yusuf, "Model-Based Verification for AI-Enabled Cyber-Physical Systems through Guided Falsification of Temporal Logic Properties," in Proc. 2025 IEEE/ACM 4th Int. Conf. on AI Engineering—Software Engineering for AI (CAIN), 2025, doi: 10.1109/CAIN66642.2025.00044.
- [8]. S. Yaghoubi and G. Fainekos, "Gray-box adversarial testing for control systems with machine learning components," in Proc. ACM Int. Conf. Hybrid Systems: Computation and Control, 2019, pp. 179–184.

