Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 29-30, 2026

Measuring the Inference Cost of Adversarial Machine Learning Mitigations: A Case Study on Diagnostic AI Supporting Autonomous Operation

Ju Hyung Song[a*], JaeKwan Park[a], Jae-Gu Song[a], Yonggu Lee[a], Seongyeol Oh[a], Inhye Hahm[a], Young Jun Lee[a]
[a]*Korea Atomic Energy Research Institute, Daejeon, South Korea*
[*]Corresponding Author: jhsong@kaeri.re.kr

***Keywords :** Adversarial machine learning, Small modular reactor, Autonomous operation, Inference cost

1. Introduction

As autonomous operation is discussed for small modular reactors (SMRs), artificial intelligence (AI) functions that diagnose and classify plant states are under consideration, with AI regarded as an enabling technology for such operation [1]. These functions, however, present an attack surface distinct from that of conventional digital assets. Attacks exploiting the statistical and data-driven nature of machine learning are collectively referred to as adversarial machine learning (AML) [2], and small signal manipulations within instrument channel uncertainty can cause an accident state to be misclassified as normal.

Mitigations recommended against such evasion attacks, including randomized smoothing, input restoration, and ensemble methods [2], share one structure: they evaluate the model multiple times for a single input and aggregate the results, so the strength of the defense is purchased with computation per inference. For a real-time diagnostic function, this cost means response delay and resource consumption, a matter of availability, and the amplified computation is triggered by any query an attacker sends, so a defense introduced for integrity creates a new availability burden. In a nuclear facility, where digital systems operate in fixed and qualified configurations, this cost cannot simply be absorbed by adding resources after deployment; whether the margin for it exists must therefore be established beforehand. The NIST taxonomy, meanwhile, leaves mitigations for some availability-related attack classes not specified, a representative example being the energy-latency attack driven merely by repeated queries [2], and the evasion mitigations above have usually been evaluated by their defensive effect, with the burden they themselves induce receiving comparatively little attention.

This study measures the defensive effect and the inference cost of AML mitigations together. The target is a diagnostic function of the kind considered for SMR autonomous operation, emulated with openly available simulation data. The results should be read as a demonstration of the measurement to be performed before deployment, not as a validation of any specific design.

2. Measurement Setup

The quantity examined in this study is the computation that a deployed prediction path spends per inference once mitigations are attached to it, together with the effort an attacker must spend against that path. Three quantities were recorded: the attack success rate as the defensive effect, the per-window inference latency as the defender cost, and the number of queries consumed by the attack as the attacker cost, with the clean macro F1 tracked as a side effect of the defense. Each measurement was repeated five times.

The prediction path was assembled in four steps: starting from an unprotected classifier (C0), randomized smoothing with 25 noise copies was attached (C1), an input quantization step was inserted before the model (C2), and the whole was replicated over a three-member ensemble (C3). Model evaluations per inference grow accordingly from 1 (C0) to 25 (C1, C2) and 75 (C3); quantization adds no evaluation. Whether the measured latency actually follows these structural multipliers is examined in Section 3. The smoothing noise is drawn from a fixed seed at every call, so that a window queried again on its own receives the same noise and returns the same label.

The workload on this bench is a four-class plant-condition classification task built from the public NPPAD simulation dataset [3], with windowed multivariate signals as input. NPPAD is generated from a conventional pressurized water reactor simulator and is not specific to SMRs. It was adopted because measurement data from operating SMRs are not publicly available, and because the cost quantities examined here arise from the composition of the prediction path rather than from the plant that produced the data. Two classifiers of deliberately opposite computational character were placed on the path: a random forest (RF), whose inference proceeds by traversing decision trees, prepared in line with an earlier NRC detection study [4], and a compact one-dimensional convolutional network (CNN), whose inference is a batched tensor operation. Both fit the clean task nearly perfectly (macro F1 1.000 and 0.9996), so any degradation observed later is attributable to the attack or to the attached mitigations.

The path was probed with a black-box, decision-based attack [5] directed at the deployed path as a whole, so that every query passed through the quantization, the noisy copies and the vote, and the attack observed only the labels the path returned. This matches an adversary who has reached the signal path but holds no internal knowledge of the model or of the attached mitigations. A

gradient-based attack plays the same role for the CNN, though it carries no query count, so the query-based cost accounting concerns the RF. Perturbations were kept within 3% of each signal span, taken here as the instrument channel uncertainty. The wall-clock time of each attack run was recorded as well.

3. Results and Discussion

Table I. Inference latency by configuration (per window, average of five repetitions, ratios computed from unrounded latencies)

Config.	Evals.	RF (ms)	RF ratio	CNN (ms)	CNN ratio
C0	1	0.259	1.00	0.007	1.00
C1	25	6.543	25.31	0.851	128.9
C2	25	6.559	25.37	0.850	128.8
C3	75	19.611	75.86	2.536	384.2

As shown in Table I, the latency ratio in the RF followed the number of evaluations closely (25.31 and 75.86 times for 25 and 75 evaluations). In the CNN, latency increased by about 129 and 384 times, far exceeding the evaluations: its unprotected inference takes only 0.007 ms per window through batch computation, so the fixed cost of noise generation and result aggregation dominates the total. Estimating the cost from the structural multiplier alone can therefore underestimate it by more than a factor of five depending on the model, and the clean macro F1 of the CNN also declined slightly (0.9996 to 0.992-0.995).

Table II. Defensive effect and two-sided costs of the RF (per window, average of five repetitions)

Config.	Success rate	Attacker queries	Defender evals.	Defender ratio
C0	0.295	305	305	1.0
C1	0.000	400	10,000	32.8
C2	0.007	402	10,050	33.0
C3	0.003	404	30,300	99.3

As shown in Table II, the attack success rate dropped sharply from 0.295 to zero once randomized smoothing was applied. Since a decision-based attack estimates the boundary from repeatedly observed labels, a zero could also reflect unstable boundary estimation under a stochastic output rather than robustness. To exclude this, C1 was attacked again with one set of noise fields applied to every window in every query, whatever the size of the query batch, so that the path was a deterministic function of its input; the success rate remained zero in all five repetitions (389 queries per window). The cost, however, was not distributed symmetrically. Attacker queries grew only 1.32 times (305 to 404), whereas defender evaluations, the queries multiplied by the evaluation multiplier of the configuration, grew about 99 times (305 to 30,300). The ratio of the two, about 75, is not an independent finding: by construction it equals the evaluation multiplier of C3, since a decision-based attack treats the entire prediction path as a single function and one query to the attacker is 75 evaluations to the defender. What the measurement adds is that the multiplier was borne almost entirely by the defender rather than offset by a longer attack, as the 1.32-fold increase in attacker queries shows.

The measured asymmetry is not a ground for rejecting mitigations; what it shows is that they can be introduced without measuring whether the margin they consume exists. Three implications follow: resource margin should be evaluated under an assumed attack, since at the batch throughput of Table I the 404 queries per window correspond to about 7.9 seconds on average (19.611 ms × 404), and in the attack runs themselves, where queries arrive one at a time and the per-call overhead of the path is not amortized, the same queries occupied 218 to 272 seconds of wall-clock time per window, against a 10-second sampling interval; the cost should be measured in the deployed configuration, since its predictability depends on the model; and mitigations differ in cost efficiency, input restoration (quantization) adding no evaluations while the ensemble tripled them for a marginal gain (0.007 to 0.003).

These results outline the requirements for a protective structure that wraps the mitigated prediction path, a harness: query rate limiting to bound attacker-side queries, presented so far only as a general principle in operational knowledge bases [6, 7], and standing monitoring of latency and resource use, with each component measurable on its own. The results are based on a single advisory diagnostic function on simulated data and on an attacker who queries the deployed path but does not otherwise exploit knowledge of the mitigations, and the asymmetry was measured for a decision-based attack; the distribution may differ for methods that require no queries to the deployed system.

4. Conclusions

The mitigations controlled the evasion attack (success rate 0.295 to zero, also with the noise held fixed), but under a decision-based attack the cost shifted asymmetrically to the defender (attacker queries 1.32 times, defender evaluations about 99 times, the structural multiplier of 75 being borne almost entirely by the defender), and the latency amplification departed from the structural multiplier by more than five times for the CNN. The choice of mitigation therefore cannot be judged by the defensive effect alone and should be evaluated in the deployed configuration together with its inference cost. Future work includes implementing the harness and extending the target to functions whose outputs drive actuation.

ACKNOWLEDGEMENTS

This work was supported by the Nuclear Safety Research Program through the Regulatory Research Management Agency for SMRs (RMAS) and the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (No. 1500-1501-409).

REFERENCES

[1] A. Kim, K. Lawson-Jenkins, D. Eskins, I. Garcia, and T. Rivera, "Cybersecurity Considerations of Autonomy in Nuclear Facilities (Cybersecurity Licensing Activities)", ICONS, 2024.
[2] A. Vassilev, A. Oprea, M. Hamin, A. Fordyce, H. Anderson, and X. Davies, NIST AI 100-2e2025, Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations, NIST, U.S., 2025.
[3] B. Qi, X. Xiao, J. Liang, L. C. Po, L. Zhang, and J. Tong, "An Open Time-Series Simulated Dataset Covering Various Accidents for Nuclear Power Plants", Scientific Data, 2022.
[4] TLR-RES/DE-2024-003, Characterizing Nuclear Cybersecurity States Using Artificial Intelligence/Machine Learning: Final Report, U.S. Nuclear Regulatory Commission, 2024.
[5] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A Query-Efficient Decision-Based Attack", IEEE S&P, 2020.
[6] AML.M0004, Limit AI Service Query Volume and Rate, MITRE ATLAS, U.S., 2026.
[7] OWASP Top 10 for LLM Applications 2025, LLM10: Unbounded Consumption, OWASP Foundation, 2025.