Research Square

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

Does Bayesian Uncertainty Make Network Intrusion Detectors Robust to Adversarial Evasion?

Mohammad Hamdan

Mohammad.Hamdan@hiast.edu.sy

Higher Institute for Applied Sciences and Technology (HIAST)

Mohammed Assora

Higher Institute for Applied Sciences and Technology (HIAST)

Mustapha Dakkak

Higher Institute for Applied Sciences and Technology (HIAST)

Research Article

Keywords: Adversarial robustness, Bayesian neural networks, Evasion attacks, Hierarchical priors, Intrusion detection, Uncertainty quantification

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11738416/v1

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

Additional Declarations: No competing interests reported.

Abstract

Bayesian neural networks are often proposed for intrusion detection because their predictive uncertainty is expected to make them harder to fool and to flag manipulated traffic. We test this expectation under an evaluation designed to avoid the pitfalls that have inflated earlier robustness claims: a leakage-aware split of de-duplicated flows, feature-space constraints that keep perturbed flows consistent, adaptive white-box attacks with expectation over transformation, a gradient-free attack, and a targeted evasion threat in which an attacker disguises its own traffic as benign. On CIC-IDS2017 and UNSW-NB15 we compare a static Bayesian neural network, two variants of a hierarchical automatic-relevance-determination prior, Monte Carlo dropout, a deep ensemble and adversarial training. Without adversarial training, every model is evaded on 47–95% of the attack flows it detects, and its worst-case macro-F1 stays below 0.21. Hierarchical priors improve worst-case macro-F1 by about 0.05 over a static Bayesian network on UNSW-NB15, but have no measurable effect on CIC-IDS2017. Uncertainty separates attacked from clean flows with an area under the receiver operating characteristic curve of up to 0.86 on CIC-IDS2017, yet an attacker aware of the detector still evades it for 38–44% of attack flows, and the signal is close to uninformative on UNSW-NB15. Adversarial training is the only effective defense, but a gradient-free attack reveals 6.0% residual evasion that gradient-based attacks alone report as 0.3%, concentrated in the rarest attack classes. We release the code to support reproducible robustness evaluation of intrusion detectors.

1 Introduction

Rule-based intrusion detection systems (IDSs) can no longer keep pace with the volume and variety of network attacks, and modern detectors increasingly rely on deep learning to recognize both known and novel attack traffic. Deep networks, however, are vulnerable to adversarial examples: small, deliberately chosen changes to the input that flip the prediction (Szegedy et al. 2014; Goodfellow et al. 2015). For an IDS this is not a theoretical concern. An attacker controls much of its own traffic and can pad packets, add packets or delay them so that a malicious flow looks benign. Such perturbations remain effective under the constraints of real protocols (Apruzzese et al. 2018; Rigaki and Garcia 2018; Sheatsley et al. 2021), and surveys of the field find that most published detectors are evaluated only on clean data and offer no protection even against simple gradient-based attacks (Alotaibi and Rassam 2023).

Bayesian neural networks (BNNs) are a natural candidate for a more robust detector (MacKay 1992; Neal 1996; Blundell et al. 2015; Gal and Ghahramani 2016). Instead of a single weight vector they learn a distribution over weights, and they report how uncertain each prediction is. Two expectations follow. First, averaging predictions over the posterior should smooth the loss surface that gradient-based attacks exploit (Carbone et al. 2020). Second, the predictive uncertainty should rise on manipulated inputs, so that it can be used to flag them (Ye and Zhu 2018). Hierarchical priors, which learn how strongly each group of weights should be regularized, extend this idea: automatic relevance determination (ARD) can shrink the weights of uninformative or attacker-controllable inputs and so, in principle, deny an attacker the features it would need to manipulate (MacKay 1992; Neal 1996).

Recent work has tempered these expectations. Once implementation errors in earlier evaluations were corrected, BNNs turned out to be far less robust, and their uncertainty far less useful for detection, than had been reported (Feng et al. 2024; Bortolussi et al. 2025). This echoes a broader lesson of adversarial machine learning: defenses that appear robust often owe their success to attacks that were not adapted to them, for example because the defense masks the gradients that the attack relies on (Athalye et al. 2018; Carlini et al. 2019; Tramèr et al. 2020). Intrusion detection adds pitfalls of its own. Widely used datasets contain labelling errors and many duplicate flows (Engelen et al. 2021), and random splits that leave near-identical flows in both training and test data, together with test sets that are rebalanced or preprocessed with statistics from the whole dataset, inflate reported scores (Arp et al. 2022).

This paper asks whether Bayesian uncertainty makes neural network intrusion detectors robust when these pitfalls are avoided. We study four questions:

- **RQ1.** Do Bayesian models, and hierarchical priors in particular, resist adaptive white-box, black-box and transfer attacks better than non-Bayesian models with uncertainty estimates (Monte Carlo (MC) dropout, deep ensembles)?
- **RQ2.** Do they prevent evasion, the realistic goal of an intruder, who perturbs its own malicious traffic so that it is classified as benign?
- **RQ3.** Is their predictive uncertainty a reliable signal for detecting attacked flows, including against an attacker who knows the detector?
- **RQ4.** How do they compare with, and combine with, adversarial training (Madry et al. 2018), the reference defense against adversarial examples?

The contributions of this paper are as follows:

1. **A leakage-aware, constraint-aware evaluation protocol for intrusion detectors.** It de-duplicates flows before splitting, fits all preprocessing on the training split, keeps the natural class distribution in the test set, and perturbs only attacker-controllable features through a projection that keeps derived features consistent and leaves clean flows unchanged. Attacks include white-box attacks with expectation over transformation, a gradient-free score-based attack, transfer, a detector-aware attack and targeted evasion, with a convergence check and paired statistics over seeds.

2. **A systematic comparison of Bayesian and non-Bayesian uncertainty against adversarial training** on CIC-IDS2017 and UNSW-NB15. It covers a static BNN, a three-level hierarchical prior with uniform and data-informed hyperpriors, a label-free online update of that prior, MC dropout, deep ensembles, adversarial training, and adversarial training combined with the hierarchical prior.

3. **Evidence on each research question.** Without adversarial training every model is evaded on 47–95% of the attack flows it detects. Hierarchical priors give a modest gain on UNSW-NB15 and none on CIC-IDS2017. Uncertainty is a moderate detection signal that an adaptive attacker bypasses. Adversarial training is the only effective defense, and the hierarchical prior adds nothing significant to it.

4. **A methodological warning.** Under feature constraints, gradient-based attacks overstate the robustness of adversarially trained detectors: on CIC-IDS2017 a gradient-free attack raises the evasion rate of an adversarially trained network from 0.3% to 6.0%, concentrated in the rarest classes.

The code and the exact command lines are released to make every number in this paper reproducible.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the models, Section 4 the threat model and the attacks, and Section 5 the experimental setup. Section 6 reports the results, Section 7 discusses them and their limitations, and Section 8 concludes.

2 Related Work

2.1 Adversarial Machine Learning in Network Intrusion Detection

Adversarial examples were first described in computer vision, where small, carefully chosen perturbations mislead state-of-the-art image classifiers (Szegedy et al. 2014; Goodfellow et al. 2015). The study that introduced the fast gradient sign method (FGSM) attributed this weakness to the locally linear behavior of neural networks rather than to overfitting (Goodfellow et al. 2015). Stronger attacks followed: projected gradient descent (PGD) (Madry et al. 2018), the optimization-based Carlini–Wagner attack (Carlini and Wagner 2017), and black-box attacks that rely only on queries (Chen et al. 2017) or on transfer from a surrogate model (Papernot et al. 2017).

In network security, perturbations must respect protocol constraints such as non-negative packet sizes and valid flag combinations, and several studies show that effective attacks remain possible under these restrictions. FGSM and PGD reduced the detection rate of a random-forest IDS on CIC-IDS2017 from 95% to below 40% (Apruzzese et al. 2018). A

generative adversarial network produced realistic malicious traffic that evades detection while preserving its function (Rigaki and Garcia 2018). Constraint-aware attacks that follow protocol specifications showed that domain constraints alone do not remove the vulnerability (Sheatsley et al. 2021). McCarthy (2023) found that robustness depends heavily on the base classifier: ensembles such as random forests and LightGBM lost 6–16% of their F1-score under attack, whereas single decision trees lost up to 90%.

Surveys of the field reach a consistent conclusion. A review of 393 IDS studies published between 2018 and 2023 found that adversarial robustness was rarely tested (Issa et al. 2024), a survey of adversarial attacks on IDSs reported that most published detectors offer no protection even against simple gradient-based perturbations (Alotaibi and Rassam 2023), and a broader review of machine-learning and generative-AI defenses noted that their adversarial vulnerability remains largely unresolved (Ankalaki et al. 2025).

2.2 Bayesian Neural Networks, Uncertainty and Robustness

Neal (1996) showed that infinitely wide BNNs converge to Gaussian processes, which gives them a principled representation of uncertainty. Blundell et al. (2015) made variational inference in BNNs practical with Bayes by Backprop, the basis of the models in this paper. Two simpler approximations are widely used as baselines: Monte Carlo dropout, which interprets dropout at test time as approximate Bayesian inference (Gal and Ghahramani 2016), and deep ensembles, which train several networks from different initializations and average their predictions (Lakshminarayanan et al. 2017).

Early work suggested that BNNs are inherently robust. Carbone et al. (2020) argued that averaging over the posterior smooths the loss surface on which gradient-based attacks rely, Ye and Zhu (2018) found that predictive variance rises on adversarial inputs and can serve as a detection signal, and Wicker et al. (2021) studied the trade-off between robustness and clean accuracy that a larger posterior variance brings. Later work tempered these claims. Feng et al. (2024) showed that, once common implementation errors were corrected, label prediction, adversarial-example detection and semantic-shift detection in state-of-the-art BNN pipelines were all far less robust than reported. Bortolussi et al. (2025) linked the remaining vulnerability to degeneracies in the geometry of the posterior, and Wicker et al. (2023) proposed certification methods for BNNs based on bound propagation. A recent primer (Arbel et al. 2023) notes that there is still no agreed way to benchmark the robustness and calibration of BNNs across inference schemes. Our study contributes such a benchmark for intrusion detection.

2.3 Hierarchical Priors

Hierarchical priors place a distribution on the hyperparameters of the weight prior and learn them from data. Automatic relevance determination (ARD), introduced by MacKay (1992) and Neal (1996), gives each input its own prior precision, so that the weights of inputs that do not help prediction are shrunk towards zero. Chen et al. (2025) used structured priors for a related purpose, combining a regularized horseshoe prior for sparsity-driven feature selection in some layers with a Gaussian scale-mixture prior in others, and improved accuracy and calibration on standard benchmarks. Whether ARD-style shrinkage also improves adversarial robustness, by reducing reliance on features an attacker can manipulate, has not to our knowledge been tested on intrusion detection. Section 3.2 describes the three-level prior we evaluate.

2.4 Adversarial Training and Robust Evaluation

Adversarial training, which trains on adversarial examples generated during training, remains the reference defense (Madry et al. 2018). TRADES (Zhang et al. 2019) trades clean accuracy against robustness through an explicit regularize, and Rice et al. (2020) showed that adversarially trained models overfit, so that the checkpoint must be selected on robust rather than clean validation performance. In intrusion detection, Awad and Zakaria (2025) combined ensemble weighting with adversarial training on UNSW-NB15-style traffic, and Heydari and Nyarko (2025) kept a neural network IDS above 80% accuracy under PGD and FGSM through adversarial augmentation.

The credibility of any robustness claim depends on the evaluation. Athalye et al. (2018) showed that many defenses rely on obfuscated gradients and fall to attacks adapted to them, and guidelines for evaluating robustness (Carlini et al. 2019) and for building adaptive attacks (Tramèr et al. 2020) followed. AutoAttack's auto-PGD (APGD) with the margin-based difference-of-logits-ratio (DLR) loss (Croce and Hein 2020) and the gradient-free square attack (Andriushchenko et al. 2020) are now

standard checks against gradient masking. For intrusion detection, Engelen et al. (2021) documented labelling errors and duplicates in CIC-IDS2017, and Arp et al. (2022) catalogued pitfalls of machine learning in security, among them data snooping and unrealistic threat models. The evaluation in this paper follows these recommendations (Sections 4 and 5).

3 Models

3.1 Architecture and Variational Family

All networks share one architecture: a multilayer perceptron with three hidden layers of 128 ReLU units, an input dimension equal to the number of features retained after preprocessing (70 for CIC-IDS2017; 194 or 195 for UNSW-NB15, depending on the categorical values present in each training split), and a SoftMax output over the native classes of each dataset (7 for CIC-IDS2017, 10 for UNSW-NB15). In the Bayesian models every weight and bias follows a mean-field Gaussian, trained with Bayes by Backprop (Blundell et al. 2015) and the reparameterization trick (Kingma and Welling 2014):

$$q_\phi(\mathbf{w}) = \prod_i \mathcal{N}\left(w_i; \mu_i, \sigma_i^2\right), \quad \sigma_i = softplus(\rho_i)$$

The means are initialized with Kaiming-uniform initialization and the standard deviations at $\sigma = 0.007$. Predictions average the softmax over posterior samples,

$$p(y \mid \mathbf{x}) \approx \frac{1}{M} \sum_{m=1}^{M} softmax\left(f_{\mathbf{w}^{(m)}}(\mathbf{x})\right), \quad \mathbf{w}^{(m)} \sim q_\phi$$

with $M = 50$ samples at evaluation time.

3.2 Three-Level Hierarchical Prior

Level 1 (weights). Weights are partitioned into groups g. In the input layer each input feature j forms its own group, the column $W_1[:, j]$, so that the prior can shrink each feature's outgoing weights independently, as in ARD (MacKay 1992; Neal 1996). Each deeper layer forms a single group. Within a group,

$$w_{g,i} \mid \lambda_g \sim \mathcal{N}\left(0, \lambda_g^{-1}\right)$$

Biases receive a fixed N(0, 1) prior. The *Static BNN* uses this level alone, with fixed precisions $\lambda = (1.0, 1.5, 1.5, 2.0)$ for the four weight tensors.

Level 2 (hyperprior). In the hierarchical models the precisions are random, with a Gamma hyperprior (shape–rate parameterization) and a variational Gamma hyper-posterior:

$$\lambda_g \sim Gamma\left(\alpha_g, \beta_g\right), \quad q\left(\lambda_g\right) = Gamma\left(a_g, b_g\right)$$

For the deeper layers $\alpha = 2$ and $\beta = 2/\lambda$, so that the prior mean equals the Static-BNN precision. For the input features we evaluate two hyperpriors. The *uniform* hyperprior sets $\mathbb{E}[\lambda_j] = \lambda_0$ for every feature. The *data-informed* hyperprior sets it from the training split:

$$\mathbb{E}\left[\lambda_j\right] = \lambda_0 e^{\gamma_m m_j}\left(1 + \gamma_s\left(1 - s_j\right)\right), \quad \alpha_j = \alpha_0, \ \beta_j = \alpha_j / \mathbb{E}\left[\lambda_j\right]$$

where $s_j \in [0, 1]$ is the mutual information between feature j and the label, normalized by its maximum and computed on at most 50,000 training rows; $m_j = 1$ if the feature is attacker-controllable (Section 4.2) and 0 otherwise; and $\lambda_0 = 1$, $\gamma_m = \gamma_s = 1$, $\alpha_0 = 2$. The values are rescaled so that their arithmetic mean equals λ_0, so the total amount of shrinkage is the same as for the uniform hyperprior and only its placement across features changes: weakly informative and manipulable features are shrunk more. Because the gradient $\partial f / \partial x_j$ flows only through $W_1[:, j]$, stronger shrinkage of a feature's column limits the gradient an

attacker obtains along that feature. We call the resulting models *AHP (uniform)* and *AHP (data-informed)*, for adaptive hierarchical prior.

Level 3 (online update). Level 3 updates the model after deployment from traffic flagged as suspicious, without labels. It is described with the detection mechanism in Section 4.5.

3.3 Training Objective

With a random precision λ, the marginal prior on the weights is a student-t, not a Gaussian, so the Gaussian closed-form Kullback–Leibler (KL) divergence does not apply. We use the exact mean-field evidence lower bound (ELBO) terms. For a group *g*,

$$\mathbb{E}_{q(\lambda_g)}\left[KL\left(q\left(\mathbf{w}_g\right) \parallel p\left(\mathbf{w}_g \mid \lambda_g\right)\right)\right] = \frac{12}{\sum_{i \in g}\left(\mathbb{E}\left[\lambda_g\right]\left(\sigma_i^2 + \mu_i^2\right) - 1 - \log\sigma_i^2 - \mathbb{E}\left[\log\lambda_g\right]\right)}$$

$$\mathbb{E}\left[\lambda_g\right] = a_g/b_g,\ \ \mathbb{E}\left[\log\lambda_g\right] = \psi\left(a_g\right) - \log b_g$$

$$KL\left(Gamma\left(a,b\right) \parallel Gamma\left(\alpha,\beta\right)\right) = (a-\alpha)\,\psi\left(a\right) - \log\Gamma\left(a\right) + \log\Gamma\left(\alpha\right) + \alpha\left(\log b - \log\beta\right) + a\,\frac{\beta - b}{b}$$

where ψ is the digamma function. Both expressions agree with Monte Carlo estimates to within 10^{-3}. The Static BNN uses the Gaussian KL with its fixed λ, the special case of the first expression with a point-mass q(λ). The per-batch loss is

$$\mathcal{L}_t = CE_{\mathbf{c}}\left(y,\ f_{\mathbf{w}}\left(\mathbf{x}\right)\right) + \kappa_t\,\frac{KL}{N},\ \ \mathbf{w} \sim\ q_\phi$$

with one posterior sample per step, *N* the number of training flows, and class weights $c_k \propto (N/N_k)^{1/2}$ normalized to mean 1. The KL weight is annealed linearly from 0.1 to 1 over the first quarter of training:

$$\kappa_t = \min\left(1,\ 0.1 + 0.9\,t/T_a\right),\ \ T_a = \max\left(1, \lfloor 0.25\,T \rfloor\right)$$

All models use Adam (learning rate 10^{-3}), batch size 256, gradient-norm clipping at 1.0, at most $T = 40$ epochs, and early stopping with patience 15 on a stratified validation split of 10% of the training data. Normally trained models keep the checkpoint with the best clean validation macro-F1. Adversarial trained models keep the checkpoint with the best validation macro-F1 under attack (PGD with 10 steps at the training budget ε, on a fixed stratified subset of 2,000 validation flows), since selecting them on clean accuracy favors early, non-robust checkpoints (Rice et al. 2020).

4 Threat Model and Evaluation

4.1 Threat Model

All perturbations are bounded in the normalized feature space, $\|\delta\|_\infty \le \varepsilon$, where ε is a fraction of each feature's training range, and they act only on attacker-controllable features (Section 4.2). We consider five attackers:

• **A1, white-box.** Knows the architecture and the variational posterior and attacks the predictive distribution with expectation over transformation (EOT): every gradient average fresh posterior samples (Athalye et al. 2018; Feng et al. 2024).

• **A2, fully adaptive.** A1 plus knowledge of the detection threshold τ_H and of the Level-3 update rule. In the prequential protocol (Section 4.5) it crafts each perturbation against the current, already adapted model.

• **A3, score-based black-box.** Sees only the averaged class probabilities returned by the deployed model and has a query budget.

• **A4, transfer.** Crafts perturbations on a surrogate (the Standard NN) and replays them against the target.

• **A5, evasion.** The realistic goal of an intruder: perturbs malicious flows only, so that they are classified as benign.

Untargeted attacks on all classes (A1–A4) measure general robustness, including benign flows pushed into an attack class. A5 measures the threat that matters most for an IDS: attacks that become invisible.

4.2 Feature-Consistent Constraints

CIC-Flow-Meter features are coupled, and many cannot be changed by an attacker who controls only its own traffic. Each perturbed flow is mapped back to raw feature space and projected onto a consistent flow:

1. **Immutable features are restored**: destination port, protocol, backward-direction (server-to-client) statistics, the backward initial window size and one-hot categorical features.

2. **Additive-only features may only increase**: forward packet counts, forward byte totals and the flow duration. An attacker can add padding, packets or delay to its own traffic but cannot remove them.

3. **Integer features move by whole units**: the change relative to the clean value is rounded.

4. **Derived features follow their primitives**: each change by exactly the change of its defining relation (mean packet length = bytes / packets; flow bytes/s and packets/s; forward and backward packets/s = count / duration), and none is perturbed independently.

5. **All values are clipped** to the training-split bounds.

For CIC-IDS2017 this leaves 52 of the 70 features attacker-controllable; 12 are immutable and 6 are derived. The projection is idempotent and leaves clean flows unchanged, project (x, x) = x, even when the stored values of a flow are not exactly consistent with each other.

These rules are written for CIC-Flow-Meter features. UNSW-NB15 uses a different feature set, for which only two rules apply: the one-hot categorical features (proto, service, state) are immutable and integer features are rounded. This leaves 39 of its 194–195 features attacker-controllable, including server-side statistics that a real attacker controls only indirectly, and no additive-only or derived-feature rules. The UNSW-NB15 threat model is therefore more permissive, robustness on UNSW-NB15 is a conservative estimate, and absolute robustness numbers should not be compared across the two datasets.

4.3 Attacks

White-box attacks on stochastic models (BNNs, MC dropout) differentiate the log of the averaged predictive distribution, re-estimated with M fresh posterior samples at every step. Each attack keeps a continuous iterate inside the L∞ ball and evaluates the model on its projection onto the constraints of Section 4.2, so that small steps on integer features accumulate instead of being rounded away. Every attack starts from the clean flow as its current best and replaces it only with a point that scores higher, so that an attack can never turn a misclassified flow into a correct one.

• **PGD-EOT (A1)** (Madry et al. 2018): K steps of size 2.5ε/K from a uniform random start, cross-entropy loss, R restarts. For each flow we keep the restart that misclassifies it, otherwise the one with the highest loss, judged on a fresh 50-sample estimate.

• **APGD-CE and APGD-DLR (A1)** (Croce and Hein 2020): step-size-free PGD with momentum. The margin-based DLR loss is insensitive to the SoftMax saturation that can hide gradients from cross-entropy.

• **Square (A3)** (Andriushchenko et al. 2020): score-based random search that sets a random subset of controllable features to $\pm\,\varepsilon$ at each query and keeps the change if it lowers the margin of the true class.

• **Transfer (A4)**: PGD on the Standard NN, replayed against each model.

• **Detector-aware (A2)**: PGD that maximizes NLL(y | x') − 5·max(0, H(x') − τ_H + 0.05); it succeeds when the flow is misclassified and its predictive entropy stays below τ_H.

• **Targeted evasion (A5)**: PGD-EOT with the same budget on malicious flows only, maximizing log p(benign | x'). Because gradient-based attacks can be misled by masked gradients, evasion is also attempted with a targeted square attack, and a flow counts as evading if either attack makes it look benign. For the Bayesian models we also run a targeted detector-aware variant.

A flow counts as robust in the worst case only if PGD-EOT, APGD-CE, APGD-DLR and Square all fail on it. Carlini–Wagner L2 was not evaluated within our compute budget; APGD-DLR provides a second, margin-based loss.

4.4 Evaluation Budget and Convergence

The evaluation budget was fitted to a single laptop GPU (Section 5.5) and is listed in Table 1.

Table 1
Evaluation budget.

Setting	Value
Perturbation budgets ε	0.02, 0.05 (main), 0.10
PGD / APGD iterations K	50
PGD restarts R	2
EOT samples M (stochastic models)	10 (1 for deterministic models)
Posterior samples for the final prediction	50
Square queries per flow	1,000
Evaluation flows (white-box attacks)	2,000 per seed, stratified, natural class proportions, at least 20 per class
Clean metrics	full test split
Seeds	3

Convergence was checked on CIC-IDS2017, seed 0, by doubling the iterations (*K* = 100) and restarts (*R* = 4) for the Standard NN and AHP (uniform). PGD alone is not converged at the main budget: its robust accuracy fell by 3.4–7.6 points (for the Standard NN at *ε* = 0.02, from 0.525 to 0.449). The worst case over all attacks is converged: its macro-F1 changed by 0.2 points for both models (0.069 → 0.067 and 0.106 → 0.104). We therefore use worst-case macro-F1 as the robustness metric throughout. In addition, on CIC-IDS2017 the black-box Square attack was stronger than the white-box attacks against the adversarially trained models, a sign that their gradients mislead gradient-based attacks (Athalye et al. 2018); this is a second reason to rely on the worst case and the reason evasion is also measured with a gradient-free attack.

4.5 Uncertainty-Based Detection and Level-3 Adaptation

Detection. For every model we flag a flow when its predictive entropy exceeds τ_H, the 95th percentile of the entropy on clean validation flows, so that the false-alarm rate of the detector is fixed at 5% and the threshold is never tuned on attacks. For deterministic models this is the softmax entropy. For the stochastic models and the deep ensemble we also use the mutual information, which isolates epistemic uncertainty:

$$H\left[p\left(y \mid \mathbf{x}\right)\right] = -\sum_{c} \bar{p}_c \log \bar{p}_c, \quad MI\left(\mathbf{x}\right) = H\left[\bar{p}\right] - \frac{1}{M} \sum_{m=1}^{M} H\left[p^{(m)}\right]$$

Level 3. Flagged flows are collected in a buffer. When it holds *B* = 128 flows, the model takes 10 Adam steps (learning rate 10^{-4}) on

$$\mathcal{L}_3 = CE_{\mathbf{c}}\left(\mathcal{R}\right) + \frac{KL}{N} + w_s \, \mathbb{E}_{\mathbf{x} \in \mathcal{B}} \, KL\left(p\left(y \mid \mathbf{x}\right) \;\middle\|\; p\left(y \mid \mathbf{x} + \mathbf{r}_{\mathbf{x}}\right)\right)$$

where R is a replay batch of 256 labelled training flows, $\mathbf{r}_x$ a one-step virtual-adversarial direction of size ε (Miyato et al. 2019), the predictive distributions use $M = 5$ samples, and $w_s = 1$. The stability term asks the prediction to be locally constant around suspicious flows without stating their label: no test labels are used. Afterwards the Level-2 hyperprior moves towards the current hyper-posterior, $\alpha \leftarrow (1 - \rho)\alpha + \rho a$ and $\beta \leftarrow (1 - \rho)\beta + \rho b$, with $\rho = 0.1$.

Proposition 1

Changing the hyperparameters (α, β) or the hyper-posterior $q(\lambda)$ alone does not change any prediction. *Proof*: $p(y \mid x)$ depends on the model only through $q_\varphi(w)$; the hyperparameters enter the ELBO only through the KL terms, so they affect predictions only through a subsequent update of φ. ■ Any online adaptation that is to change predictions must therefore update φ, which is why the update above includes the posterior.

Prequential protocol. Level 3 is evaluated predict-then-adapt. A hold-out of 2,000 test flows is set aside and a stream of 5,000 flows is processed in chunks of 1,024. In each chunk 30% of the flows are replaced by adversarial versions crafted against the current, already adapted model with PGD-EOT (25 steps) or the detector-aware attack. The model predicts the whole chunk before it may adapt on the flagged flows. After the stream, the hold-out is attacked afresh against the adapted model.

5 Experimental Setup

5.1 Datasets

We use CIC-IDS2017 (Sharafaldin et al. 2018) and UNSW-NB15 (Moustafa and Slay 2015) with their native attack categories; no class is remapped to another taxonomy.

• **CIC-IDS2017**: Benign, DoS (Hulk, GoldenEye, Slowloris, Slowhttptest), DDoS, PortScan, BruteForce (FTP-Patator, SSH-Patator), WebAttack (brute force, XSS, SQL injection) and Bot. Infiltration and Heartbleed have fewer than 50 flows after de-duplication and are excluded.

• **UNSW-NB15**: Normal and the nine attack categories of the attack_cat field (10 classes). The official training and testing partitions are merged and re-split with the same protocol as CIC-IDS2017, so that both datasets are treated identically.

5.2 Leakage-Aware Preprocessing

The order of operations prevents any test-set statistic from reaching the model:

1. Labels are mapped to native categories.

2. Identifier columns are dropped (flow ID, source and destination IP, source port and timestamp; for UNSW-NB15 also srcip, dstip, sport, stime, ltime and id). The destination port is kept as an immutable feature.

3. Exact duplicate rows (features and label) are removed, since CIC flow exports contain many duplicates (Engelen et al. 2021) and duplicates shared between training and test data inflate scores.

4. A stratified split puts 20% of the flows in the test set; 10% of the remainder forms a stratified validation set.

5. All preprocessing is fitted on the training split only and then applied to validation and test data: infinite values are imputed with the training median, zero-variance features are dropped, categorical features (proto, service, state) are one-hot encoded with the training vocabulary, and features are min–max scaled between the training minimum and the training 99.9th percentile and clipped to [0, 1].

6. Class imbalance is handled on the training split only: each class is capped at 50,000 training flows by random sampling, and the loss is class-weighted (Section 3.3). Validation and test data keep their natural class distribution, so that reported scores reflect realistic base rates.

Because much prior work balances or scales the data before splitting, clean scores here, in particular macro-F1 on rare classes, are lower than the figures often reported for these datasets and are not directly comparable to them.

5.3 Models Compared

All eight models share the architecture of Section 3.1 and the optimization of Section 3.3:

1. **Standard NN**: deterministic network.

2. **MC Dropout** (Gal and Ghahramani 2016): dropout p = 0.1 after each hidden layer, kept active at test time, 50 samples per prediction.

3. **Deep Ensemble** (Lakshminarayanan et al. 2017): five Standard NNs trained from different initializations, with averaged class probabilities.

4. **PGD-AT** (Madry et al. 2018): the Standard NN trained only on adversarial examples generated at every batch by constrained PGD (10 steps, $\varepsilon = 0.05$), with robust checkpoint selection.

5. **Static BNN**: Level 1 only, fixed precisions.

6. **AHP (uniform)**: Levels 1–2, uniform hyperprior.

7. **AHP (data-informed)**: Levels 1–2, data-informed hyperprior.

8. **AHP (uniform) + PGD-AT**: model 6 trained with the adversarial training of model 4 on the ELBO loss. It tests whether the hierarchical prior adds anything on top of the reference defense.

Level 3 is evaluated separately, in the prequential protocol, for models 6 and 7. We also trained TRADES (Zhang et al. 2019) with $\beta = 6$ and $\beta = 1$. On CIC-IDS2017 it did not learn robustness under our feature constraints: its validation macro-F1 under PGD fell from about 0.65 in the first epoch to 0.13 by the tenth, for both values of β. Since a baseline that does not train properly would make the comparison unfair, we report PGD-AT as the adversarial-training baseline and leave a constraint-aware TRADES to future work.

5.4 Metrics and Statistical Analysis

Classification: macro-F1 (primary metric, since it weights every class equally under strong imbalance), accuracy and per-class F1. **Calibration**: expected calibration error (ECE) with 15 equal-width bins. **Robustness**: macro-F1 under each attack and in the worst case over all attacks; following the standard convention, a flow counts as correct under an attack only if it is classified correctly both on clean data and after the attack. **Evasion (A5)**: among the malicious flows a model detects on clean data, the fraction classified as benign after the attack, and the fraction that is also below τ_H. **Detection**: area under the receiver operating characteristic curve (AUROC) and true-positive rate at 5% false-positive rate for entropy and mutual information.

Each seed (0, 1, 2) changes the data split, the initialization and the evaluation subset. Tables report mean ± standard deviation over seeds. Comparisons between two models use the per-seed paired difference, with a 95% confidence interval from the t-distribution and a paired t-test; we claim an improvement only when the interval excludes zero. With three seeds these intervals are wide, so an effect we do not detect is not evidence of its absence.

5.5 Hardware and Software

All experiments ran on a single laptop with an Intel Core i7-13650HX processor, 24 GB of RAM and an NVIDIA GeForce RTX 3050 Laptop GPU with 6 GB of memory, under Windows 11 Pro, with Python 3.11.9, PyTorch 2.14.1 (CUDA 12.6) and scikit-learn 1.9.1. The evaluation budget in Table 1 was chosen to fit this hardware. Figures were produced with Matplotlib.

5.6 Use of AI Tools

An AI assistant (Claude, Anthropic) was used to help develop and review the experimental code, rephrase sentences and correct grammar. The authors reviewed and verified all code, results and text, and take full responsibility for the content of this article.

6 Results

All numbers are means over three seeds, with standard deviations where shown. Robustness is the worst-case macro-F1 at $\varepsilon = 0.05$ (Section 4.4).

6.1 Clean Performance

Table 2 reports performance on the full, unattacked test splits.

Table 2
Clean performance on the full test splits (mean ± SD over three seeds).

Model	CIC-IDS2017 accuracy	CIC-IDS2017 macro-F1	CIC-IDS2017 ECE	UNSW-NB15 accuracy	UNSW-NB15 macro-F1	UNSW-NB15 ECE
Standard NN	0.987 ± 0.003	0.778 ± 0.007	0.006 ± 0.004	0.757 ± 0.007	0.534 ± 0.010	0.031 ± 0.013
MC Dropout	0.986 ± 0.004	0.781 ± 0.012	0.007 ± 0.002	0.762 ± 0.007	0.537 ± 0.015	0.028 ± 0.006
Deep Ensemble	0.987 ± 0.001	0.783 ± 0.001	0.006 ± 0.001	0.772 ± 0.002	0.553 ± 0.005	0.021 ± 0.002
PGD-AT	0.976 ± 0.002	0.756 ± 0.004	0.014 ± 0.003	0.661 ± 0.009	0.399 ± 0.012	0.031 ± 0.011
Static BNN	0.986 ± 0.003	0.775 ± 0.007	0.005 ± 0.001	0.743 ± 0.007	0.532 ± 0.012	0.032 ± 0.007
AHP (uniform)	0.983 ± 0.003	0.766 ± 0.011	0.005 ± 0.002	0.752 ± 0.009	0.510 ± 0.007	0.033 ± 0.009
AHP (data-informed)	0.984 ± 0.004	0.766 ± 0.008	0.006 ± 0.001	0.741 ± 0.008	0.497 ± 0.025	0.031 ± 0.010
AHP (uniform) + PGD-AT	0.945 ± 0.004	0.689 ± 0.019	0.034 ± 0.003	0.670 ± 0.006	0.400 ± 0.007	0.026 ± 0.001

On CIC-IDS2017, accuracy is close to 0.99 for every normally trained model, but macro-F1 is below 0.79. The gap comes from the rarest classes (Bot and Web Attack), which together make up well under 1% of the test flows under the natural class distribution we keep. UNSW-NB15 is harder: its ten classes include several (Analysis, Backdoor, DoS, Exploits) that overlap strongly in feature space, and macro-F1 lies between 0.50 and 0.55.

Without adversarial training, the Bayesian models cost little clean performance: AHP (uniform) differs from the Static BNN by − 0.009 macro-F1 on CIC-IDS2017 (95% CI [− 0.041, + 0.023]) and − 0.022 on UNSW-NB15 ([− 0.066, + 0.021]). The Deep Ensemble has the best clean macro-F1 on both datasets. Adversarial training has a cost that depends strongly on the dataset: PGD-AT loses 0.021 macro-F1 relative to the Standard NN on CIC-IDS2017 ([− 0.040, − 0.003]) but 0.136 on UNSW-NB15 ([− 0.160, − 0.112]). Adding the hierarchical prior to adversarial training costs a further 0.068 on CIC-IDS2017 ([− 0.123, − 0.012]) and nothing measurable on UNSW-NB15 (+ 0.001, [− 0.043, + 0.045]).

6.2 Robustness under Attack

Table 3 reports robustness at $\varepsilon = 0.05$ as the worst-case macro-F1, together with macro-F1 under PGD-EOT alone and the calibration error under PGD.

Table 3
Robustness at $\varepsilon = 0.05$ (mean ± SD over three seeds). Worst case: correct on clean data and under every attack. PGD only: an upper bound on robustness, for reference.

Model	CIC-IDS2017 worst-case macro-F1	CIC-IDS2017 PGD-only macro-F1	CIC-IDS2017 ECE under PGD	UNSW-NB15 worst-case macro-F1	UNSW-NB15 PGD-only macro-F1	UNSW-NB15 ECE under PGD
Standard NN	0.056 ± 0.014	0.082	0.654	0.053 ± 0.010	0.059	0.529
MC Dropout	0.104 ± 0.031	0.126	0.375	0.128 ± 0.019	0.145	0.333
Deep Ensemble	0.093 ± 0.031	0.125	0.376	0.085 ± 0.019	0.096	0.357
PGD-AT	0.723 ± 0.093	0.896	0.016	0.382 ± 0.008	0.382	0.042
Static BNN	0.095 ± 0.008	0.123	0.316	0.151 ± 0.009	0.169	0.315
AHP (uniform)	0.094 ± 0.023	0.108	0.343	0.201 ± 0.009	0.215	0.264
AHP (data-informed)	0.083 ± 0.011	0.119	0.336	0.198 ± 0.003	0.205	0.261
AHP (uniform) + PGD-AT	0.809 ± 0.032	0.833	0.031	0.350 ± 0.018	0.369	0.048

Without adversarial training no model keeps a worst-case macro-F1 above 0.21 on either dataset (Fig. 1). On CIC-IDS2017 almost every attack class is misclassified under PGD (per-class F1 at most 0.15, for DoS, and near zero for the others), and what remains of macro-F1 comes from benign traffic. On UNSW-NB15 the Bayesian models keep part of one attack class: under PGD the F1 of Generic is 0.829 ± 0.049 for AHP (uniform) and 0.809 ± 0.040 for AHP (data-informed), against 0.533 for the Static BNN, 0.465 for MC Dropout and 0.047 for the Standard NN. This single class accounts for much of the hierarchical models' advantage on that dataset (Section 6.4). Uncertainty-aware models reduce the calibration error under attack by a third to a half relative to the Standard NN, but better calibration does not translate into correct predictions.

Adversarial training changes the picture. PGD-AT raises worst-case macro-F1 over the Standard NN by 0.668 on CIC-IDS2017 (95% CI [+ 0.420, + 0.915]) and by 0.329 on UNSW-NB15 ([+ 0.286, + 0.372]), and keeps the calibration error under attack below 0.05. The difference between the two robustness columns matters for adversarially trained models: on CIC-IDS2017, PGD-AT reaches 0.896 under PGD alone but 0.723 in the worst case, because the gradient-free Square attack succeeds where the gradient-based attacks fail. Reporting PGD alone would overstate its robustness by 0.17 macro-F1. On UNSW-NB15 the two columns agree for PGD-AT.

6.3 Evasion

Table 4 reports the evasion rate of each model under the targeted attacks of attacker A5 (Section 4.1).

Table 4
Evasion rate (%) at $\varepsilon = 0.05$: among malicious flows detected on clean data, the share classified as benign after a targeted attack. Mean over three seeds; worst case with SD.

Model	CIC-IDS2017 PGD	CIC-IDS2017 Square	CIC-IDS2017 worst case	UNSW-NB15 PGD	UNSW-NB15 Square	UNSW-NB15 worst case
Standard NN	83.1	92.9	92.9 ± 4.3	79.4	80.3	81.2 ± 2.4
MC Dropout	78.8	66.0	79.6 ± 10.4	65.4	43.7	66.3 ± 4.7
Deep Ensemble	87.8	91.4	91.5 ± 3.8	70.7	73.0	74.1 ± 2.6
PGD-AT	0.3	6.0	6.0 ± 4.1	0.0	0.0	0.0 ± 0.1
Static BNN	87.1	69.2	87.7 ± 3.7	55.2	32.5	55.4 ± 4.6
AHP (uniform)	95.0	74.0	95.1 ± 2.6	50.4	30.5	50.5 ± 0.4
AHP (data-informed)	79.1	63.7	79.6 ± 4.2	46.1	28.8	47.2 ± 8.0
AHP (uniform) + PGD-AT	0.3	0.3	0.5 ± 0.9	0.0	0.0	0.0 ± 0.1

Without adversarial training an attacker evades detection for most attack flows (Fig. 2): between 79.6% and 95.1% on CIC-IDS2017 and between 47.2% and 81.2% on UNSW-NB15. On UNSW-NB15 the Bayesian models are evaded less often than the Standard NN, MC Dropout and the Deep Ensemble; on CIC-IDS2017 no uncertainty-aware model improves on the Standard NN by more than the seed-to-seed variation.

Adversarial training reduces evasion sharply, by 86.8 percentage points on CIC-IDS2017 (95% CI [− 91.7, − 82.0]) and 81.1 on UNSW-NB15 ([− 87.2, − 75.1]) relative to the Standard NN, but it does not eliminate it. On CIC-IDS2017, PGD-AT still lets 6.0% of detected attack flows evade, almost all of them found only by the gradient-free attack (0.3% with PGD alone). The residual evasion is concentrated in the rarest classes: 31.7 ± 22.5% of Bot flows, 36.8 ± 54.9% of WebAttack flows and 11.1 ± 19.2% of PortScan flows, against at most 1.7% for the frequent classes. An evaluation based on PGD alone would have reported adversarial training as close to perfect against evasion. With the hierarchical prior added, worst-case evasion on CIC-IDS2017 falls from 6.0% to 0.5%, but this difference is not significant with three seeds (− 5.5 points, 95% CI [− 18.0, + 7.0]).

For the stochastic models Square evades less often than PGD, plausibly because the score-based search sees probabilities averaged over a finite number of posterior samples, whose noise can mislead a random search. For these models PGD gives the binding rate; for the deterministic and adversarially trained models, Square does.

6.4 Does the Hierarchical Prior Help?

Table 5 gives the paired differences that answer this question directly.

Table 5
Paired differences A − B over three seeds, mean [95% CI]. Positive worst-case macro-F1 and negative evasion favor A. Bold: interval excludes zero.

A vs B	CIC-IDS2017 Δ worst-case macro-F1	CIC-IDS2017 Δ evasion	UNSW-NB15 Δ worst-case macro-F1	UNSW-NB15 Δ evasion
AHP (uniform) vs Static BNN	−0.001 [− 0.037, + 0.035]	+ 0.074 [− 0.081, + 0.229]	**+ 0.050 [+ 0.022, + 0.077]**	−0.050 [− 0.172, + 0.072]
AHP (data-informed) vs Static BNN	−0.012 [− 0.034, + 0.010]	−0.081 [− 0.245, + 0.084]	**+ 0.046 [+ 0.026, + 0.067]**	−0.083 [− 0.186, + 0.020]
AHP (data-informed) vs AHP (uniform)	−0.011 [− 0.052, + 0.030]	**−0.155 [− 0.229, − 0.081]**	−0.003 [− 0.033, + 0.026]	−0.033 [− 0.240, + 0.174]
AHP (uniform) vs MC Dropout	−0.010 [− 0.033, + 0.013]	+ 0.155 [− 0.038, + 0.347]	**+ 0.073 [+ 0.025, + 0.121]**	**−0.158 [− 0.283, − 0.033]**
AHP (uniform) vs Deep Ensemble	+ 0.001 [− 0.097, + 0.099]	+ 0.036 [− 0.122, + 0.193]	**+ 0.116 [+ 0.064, + 0.168]**	**−0.236 [− 0.308, − 0.164]**
Static BNN vs Deep Ensemble	+ 0.002 [− 0.077, + 0.081]	−0.038 [− 0.101, + 0.024]	**+ 0.067 [+ 0.040, + 0.093]**	**−0.186 [− 0.236, − 0.136]**
AHP (uniform) + PGD-AT vs PGD-AT	+ 0.086 [− 0.067, + 0.239]	−0.055 [− 0.180, + 0.070]	−0.032 [− 0.091, + 0.027]	+ 0.000 [− 0.000, + 0.000]

The answer depends on the dataset. **On CIC-IDS2017 the hierarchical prior has no measurable effect**: every interval for AHP against the Static BNN, MC Dropout and the Deep Ensemble includes zero. The one significant difference is between the two AHP variants, where the data-informed hyperprior lowers evasion by 15.5 points without changing worst-case macro-F1.

On UNSW-NB15 it gives a modest but consistent gain. Both AHP variants improve worst-case macro-F1 over the Static BNN by about 0.05, and AHP (uniform) improves on MC Dropout by 0.07 and on the Deep Ensemble by 0.12, while also lowering evasion relative to both. Part of this gain is Bayesian rather than hierarchical, since the Static BNN itself beats the Deep Ensemble by 0.067; the hierarchical prior adds roughly 0.05 on top. Most of this margin comes from one attack class, Generic (Section 6.2). The data-informed hyperprior adds nothing over the uniform one in worst-case macro-F1 on either dataset.

On top of adversarial training the prior adds nothing significant. On CIC-IDS2017, AHP (uniform) + PGD-AT has a higher mean worst-case macro-F1 than PGD-AT (+ 0.086) and lower evasion (− 5.5 points), but both intervals include zero, and it costs 0.068 clean macro-F1 (Section 6.1). On UNSW-NB15 the two are indistinguishable.

These comparisons use three seeds, and Table 5 is a selection from a larger set of paired tests. The UNSW-NB15 effects point the same way in every comparison involving a hierarchical model, but individual p-values between 0.005 and 0.03 would not all survive a family-wise correction over every comparison we ran. We therefore read them as consistent evidence of a modest effect, not as a definitive finding.

6.5 Uncertainty and Detection

Table 6 reports how well predictive uncertainty separates clean from PGD-perturbed flows, and how much evasion escapes the entropy detector.

Table 6

Detection and posterior diagnostics (mean over three seeds). AUROC: clean vs PGD-perturbed flows, scored by predictive entropy (H) or mutual information (MI). Undet.: worst-case evasion with entropy below τ_H (%). Adapt.: detector-aware targeted attack that also keeps entropy below τ_H (%; Bayesian models only). $\sigma/|\mu|$: median ratio for the first-layer weights.

Model	CIC AUROC (H)	CIC AUROC (MI)	CIC undet. (%)	CIC adapt. (%)	UNSW AUROC (H)	UNSW AUROC (MI)	UNSW undet. (%)	UNSW adapt. (%)	σ/|μ| CIC / UNSW
Standard NN	0.707	–	82.9	–	0.551	–	80.8	–	–
MC Dropout	0.802	0.790	64.2	–	0.566	0.741	66.1	–	–
Deep Ensemble	0.817	0.829	71.2	–	0.631	0.849	73.5	–	–
PGD-AT	0.527	–	0.0	–	0.566	–	0.0	–	–
Static BNN	0.843	0.851	71.0	41.2	0.597	0.736	55.4	24.8	1.34 / 6.76
AHP (uniform)	0.850	0.857	75.6	44.4	0.573	0.716	50.5	28.2	2.27 / 13.73
AHP (data-informed)	0.849	0.857	59.8	38.5	0.572	0.680	47.2	25.4	2.70 / 13.32
AHP (uniform) + PGD-AT	0.550	0.560	0.0	0.0	0.553	0.582	0.0	0.0	1.06 / 19.36

The Bayesian posteriors are genuinely dispersed. The median ratio $\sigma/|\mu|$ of the first-layer weights is above 1 for every BNN on both datasets, and the mutual information rises under attack (on CIC-IDS2017, from 0.005–0.007 on clean flows to 0.10–0.17 under PGD). The models are not regularized deterministic networks in disguise.

On CIC-IDS2017 uncertainty is a useful but limited detection signal. The BNNs separate attacked from clean flows with AUROC 0.84–0.86, above the Deep Ensemble (0.82), MC Dropout (0.80) and the SoftMax entropy of the Standard NN (0.71), with no clear advantage of the hierarchical prior. Detection does not stop evasion, however: 60–76% of detected attack flows still evade with an entropy low enough to pass the detector, and an attacker that explicitly keeps entropy below the threshold still succeeds on 38–44% of them.

On UNSW-NB15 the entropy detector is close to useless. Its AUROC lies between 0.55 and 0.63 for every model. Clean UNSW-NB15 predictions are already so uncertain that τ_H lies between 1.70 and 1.75 nats, close to the maximum of $\ln 10 \approx 2.30$ for ten classes, so attacks barely raise entropy above it. Mutual information, which isolates epistemic uncertainty, does better (0.68–0.85), and the Deep Ensemble gives the best separation (0.849).

At the operating point, the ranking changes. Figure 3 shows the full ROC curves of the uncertainty scores, which reveal how detection behaves at the low false-positive rates an IDS can tolerate. At 5% false positives, on CIC-IDS2017 the mutual information of the Deep Ensemble flags 60.7 ± 4.9% of attacked flows and that of the Static BNN 59.2 ± 4.5%, against 48.7 ± 5.0% for AHP (uniform), 42.2 ± 14.7% for MC Dropout and 26.0 ± 11.0% for the SoftMax entropy of the Standard NN. The higher AUROC of AHP (uniform) comes from false-positive rates above 20%, which no IDS would accept. On UNSW-NB15 the Deep Ensemble dominates at every operating point (52.2 ± 7.3% at 5% false positives), followed by MC Dropout (30.3%), the Static BNN (27.2%) and AHP (uniform) (16.5%), while the Standard NN is at chance level (3.0%).

The curves in Fig. 3 were obtained by repeating the attack, and the mutual information of the BNNs is estimated from ten posterior samples. Repeating attack and sampling changed the AUROC of the Standard NN, MC Dropout and the Deep Ensemble by at most 0.006, but that of the BNNs by up to 0.07, most for AHP (uniform) on UNSW-NB15, whose posterior is the most dispersed. The detection results for the BNNs in Table 6 are therefore accurate only to a few hundredths.

Adversarial training removes the uncertainty signal. For the adversarially trained models, detection AUROC falls to 0.53–0.58 on both datasets, and for AHP (uniform) + PGD-AT the mutual information under attack is only 0.006–0.007. These models do not need the detector, since evasion is already low, but they no longer offer it either.

6.6 Level-3 Online Adaptation

Table 7 compares the two AHP models with and without the label-free Level-3 update, on hold-out flows attacked afresh after the stream.

Table 7
Hold-out accuracy after the prequential stream (30% adversarial flows), mean ± SD over three seeds.

Model	Stream attack	CIC-IDS2017 none	CIC-IDS2017 full	UNSW-NB15 none	UNSW-NB15 full
AHP (uniform), clean hold-out	PGD	0.979 ± 0.004	0.977 ± 0.004	0.757 ± 0.008	0.746 ± 0.004
AHP (uniform), attacked hold-out	PGD	0.602 ± 0.098	0.584 ± 0.112	0.466 ± 0.008	0.468 ± 0.006
AHP (uniform), attacked hold-out	detector-aware	0.797 ± 0.013	0.815 ± 0.014	0.524 ± 0.013	0.518 ± 0.004
AHP (data-informed), clean hold-out	PGD	0.981 ± 0.005	0.977 ± 0.008	0.747 ± 0.007	0.738 ± 0.004
AHP (data-informed), attacked hold-out	PGD	0.555 ± 0.114	0.522 ± 0.128	0.472 ± 0.012	0.474 ± 0.009
AHP (data-informed), attacked hold-out	detector-aware	0.808 ± 0.046	0.814 ± 0.049	0.525 ± 0.013	0.527 ± 0.008

The update gives no consistent benefit. Every difference between *none* and *full* is small (at most 3.3 points) relative to the spread across seeds, and the signs disagree between attacks and datasets: on CIC-IDS2017 adaptation lowers attacked hold-out accuracy against PGD by 1.8–3.3 points and raises it against the detector-aware attack by 0.6–1.8 points. The update also triggers rarely: on average 4.7 times over a 5,000-flow stream under PGD on CIC-IDS2017, and once on UNSW-NB15, where the entropy detector flags few attacked flows. Accuracy rather than macro-F1 is reported here because a random stream of 5,000 flows rarely contains the rarest classes, which makes macro-F1 unstable.

6.7 Summary of Findings

1. **Bayesian uncertainty alone does not protect against evasion.** Without adversarial training every model is evaded on 47–95% of the attack flows it detects, and worst-case macro-F1 stays below 0.21 (RQ1, RQ2).

2. **Hierarchical priors help on one dataset only, and modestly**: about + 0.05 worst-case macro-F1 over a Static BNN and + 0.07 to + 0.12 over MC Dropout and a Deep Ensemble on UNSW-NB15, and no measurable effect on CIC-IDS2017 (RQ1).

3. **Uncertainty gives a moderate detection signal that adaptive attackers bypass**, and it is close to uninformative on UNSW-NB15 (RQ3).

4. **Adversarial training is the only effective defense**, raising worst-case macro-F1 by 0.33–0.67 and cutting evasion by more than 80 points, at a clean cost of 0.021 macro-F1 on CIC-IDS2017 and 0.136 on UNSW-NB15. The hierarchical prior adds nothing significant on top of it (RQ4).

5. **Gradient-based evaluation overstates the robustness of adversarially trained detectors**: on CIC-IDS2017 a gradient-free attack raises the evasion rate of PGD-AT from 0.3% to 6.0%, concentrated in the rarest classes.

6. **Label-free online adaptation does not improve robustness.**

7 Discussion

7.1 Why Bayesian Uncertainty Does Not Stop Evasion

The expectation that posterior averaging smooths the loss surface (Carbone et al. 2020) does not survive an attacker who averages too. With expectation over transformation, the attacker follows the gradient of the averaged prediction, which is exactly the quantity the defender reports, and the stochasticity of the posterior no longer hides the direction of attack. Our posteriors are not collapsed: $\sigma/|\mu|$ exceeds 1 and mutual information rises under attack, so the failure is not an artefact of a degenerate Bayesian model. It reflects the fact that the perturbations an intruder can make, such as adding packets, padding or delay within ε of each feature's range, move flows across the decision boundary of every network we trained. These results agree with recent findings on BNNs in vision (Feng et al. 2024; Bortolussi et al. 2025) and extend them to intrusion detection, where the attacker's room is further restricted by protocol constraints but remains sufficient.

The modest gain of the hierarchical prior on UNSW-NB15 and its absence on CIC-IDS2017 suggest a narrower role for ARD than the one that motivated it. On UNSW-NB15, with 194–195 inputs, most of them sparse one-hot indicators, the learned posterior is far more dispersed (median $\sigma/|\mu|$ 13.7 against 2.3), which is consistent with strong shrinkage of uninformative inputs, and the gain concentrates in one class. On CIC-IDS2017, whose 70 features are mostly informative flow statistics, there is little to shrink. We read this as a hypothesis, not a finding: our experiments do not isolate the feature set as the cause, and a direct test would vary the number of uninformative inputs on one dataset. The data-informed hyperprior, which shrinks attacker-controllable and weakly informative features more, did not improve worst-case macro-F1 on either dataset, so concentrating shrinkage on manipulable features is not by itself a defense.

Uncertainty is more useful as a monitoring signal than as a defense. On CIC-IDS2017 it separates attacked from clean flows with AUROC up to 0.86, yet at a 5% false-positive rate no model flags more than about 61% of attacked flows (Fig. 3), and an attacker who knows the threshold still evades for 38–44% of attack flows, and on a dataset where clean predictions are already uncertain, such as UNSW-NB15, entropy carries almost no signal. Mutual information, which separates epistemic from aleatoric uncertainty, is the more informative score there, and the deep ensemble provides it at least as well as the BNNs.

7.2 Implications for Practice and Evaluation

For practitioners, adversarial training is the defense to start from, and a Bayesian model is not a substitute for it. Its cost, however, depends strongly on the data: 0.021 clean macro-F1 on CIC-IDS2017 but 0.136 on UNSW-NB15. An operator would weigh that loss of clean detection against the near-elimination of evasion. Uncertainty estimates remain worth reporting to analysts, as long as they are not relied on to catch an adaptive attacker.

For evaluators, three practices changed our conclusions and should be routine. First, gradient-based attacks alone are not enough: under feature constraints they understated the evasion rate of adversarially trained models twentyfold (0.3% against 6.0%), and a gradient-free attack such as Square should always be included. Second, the worst case over several attacks should be the headline metric, with a convergence check; robust accuracy under PGD alone was still falling when we doubled its budget, while the worst case was stable. Third, results should be broken down by class: the residual evasion of adversarial training lives almost entirely in the rarest classes, Bot and Web Attack, which aggregate accuracy hides.

7.3 Limitations

The study has several limitations, each of which bounds its conclusions.

• **Statistical power.** Three seeds give wide confidence intervals. Effects we report as non-significant, such as the gain of AHP (uniform) + PGD-AT on CIC-IDS2017, may exist; and the UNSW-NB15 gains, while consistent, would not all survive a strict family-wise correction.

• **Datasets and threat model.** We use two datasets, and the feature constraints are fully specified only for CIC-IDS2017. On UNSW-NB15 the threat model is more permissive, so absolute robustness is not comparable across datasets. CIC-IDS2018 and more recent datasets remain to be tested.

• **Feature-space attacks.** Perturbations are consistent in feature space but are not replayed as real packets. Problem-space attacks that generate actual traffic could be more or less constrained than our projection assumes.

• **Architecture and budget.** All models are three-layer perceptron, and the evaluation budget was fitted to a single laptop GPU. The worst-case metric was converged at this budget, but PGD alone was not.

• **Baselines.** TRADES did not train properly under our constraints and is not reported, and certified defenses were not evaluated.

• **Level 3.** The negative result concerns one entropy-triggered, label-free update rule; other online adaptation schemes may behave differently.

8 Conclusion

We asked whether Bayesian uncertainty makes neural network intrusion detectors robust to adversarial evasion, under an evaluation built to avoid the leakage, unadapted attacks and gradient masking that have inflated earlier claims. On CIC-IDS2017 and UNSW-NB15 the answer is largely negative. Static and hierarchical Bayesian networks, MC dropout and deep ensembles are all evaded on most of the attack flows they detect. Hierarchical ARD priors give a modest gain on one dataset and none on the other, and add nothing significant to adversarial training. Predictive uncertainty is a moderate detection signal that an adaptive attacker bypasses and that adversarial training removes. Adversarial training is the only effective defense, and even its robustness is overstated unless gradient-free attacks are part of the evaluation.

Future work should test these conclusions with problem-space attacks that generate real traffic, on more recent datasets and with fully specified constraints for each feature set; isolate the conditions under which ARD shrinkage helps by varying the number of uninformative inputs; develop constraint-aware variants of TRADES and other adversarial-training methods for imbalanced flow data; and combine adversarial training with ensembles to restore the uncertainty signal that adversarial training alone removes.

Declarations

Funding [The authors did not receive support from any organization for the submitted work.]

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval Not applicable. The study uses publicly available network-traffic datasets and involves no human participants or animals.

Consent to participate Not applicable.

Consent for publication Not applicable.

Data availability CIC-IDS2017 and UNSW-NB15 are publicly available from their authors (Moustafa and Slay 2015; Sharafaldin et al. 2018).

Code availability The code, configuration and exact command lines that produce every table and figure are available at https://github.com/mohammademail86-del/bnn-ids-evaluation.

Author contributions [Mohammad Hamdan: conceptualization, methodology, software, investigation, formal analysis, writing – original draft. Mohammed Assora: supervision, methodology, writing – review and editing. Mustapha Dakkak: supervision, writing – review and editing.]

References

1. Alotaibi A, Rassam MA (2023) Adversarial machine learning attacks against intrusion detection systems: a survey on strategies and defense. Future Internet 15(2):62. https://doi.org/10.3390/fi15020062
2. Andriushchenko M, Croce F, Flammarion N, Hein M (2020) Square attack: a query-efficient black-box adversarial attack via random search. Computer Vision – ECCV 2020. Lecture Notes in Computer Science. Springer, Cham, pp 484–501
3. Ankalaki S, Atmakuri AR, Pallavi M, Hukkeri GS, Jan T, Naik GR (2025) Cyber attack prediction: from traditional machine learning to generative artificial intelligence. IEEE Access 13:44662–44706. https://doi.org/10.1109/ACCESS.2025.3547433
4. Apruzzese G, Colajanni M, Ferretti L, Guido A, Marchetti M (2018) On the effectiveness of machine and deep learning for cyber security. In: Proceedings of the 10th International Conference on Cyber Conflict (CyCon), pp 371–390
5. Arbel J, Pitas K, Vladimirova M, Fortuin V (2023) A primer on Bayesian neural networks: review and debates. arXiv:2309.16314
6. Arp D, Quiring E, Pendlebury F, Warnecke A, Pierazzi F, Wressnegger C, Cavallaro L, Rieck K (2022) Dos and don'ts of machine learning in computer security. In: Proceedings of the 31st USENIX Security Symposium, pp 3971–3988
7. Athalye A, Carlini N, Wagner D (2018) Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning (ICML), pp 274–283
8. Awad Z, Zakaria M (2025) An enhanced ensemble defense framework for boosting adversarial robustness of intrusion detection systems. Sci Rep 15:14177. https://doi.org/10.1038/s41598-025-94023-z
9. Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural networks. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), pp 1613–1622
10. Bortolussi L, Carbone G, Laurenti L, Patane A, Sanguinetti G, Wicker M (2025) On the robustness of Bayesian neural networks to adversarial attacks. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2024.3386642
11. Carbone G, Wicker M, Laurenti L, Patane A, Bortolussi L, Sanguinetti G (2020) Robustness of Bayesian neural networks to gradient-based attacks. Adv Neural Inf Process Syst 33(NeurIPS):15602–15613
12. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), pp 39–57
13. Carlini N, Athalye A, Papernot N, Brendel W, Rauber J, Tsipras D, Goodfellow I, Madry A, Kurakin A (2019) On evaluating adversarial robustness. arXiv:1902.06705
14. Chen P-Y, Zhang H, Sharma Y, Yi J, Hsieh C-J (2017) ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec), pp 15–26
15. Chen X, Sima L, Wei Z, Duan X, Feng P, Song F (2025) Variational sparse Bayesian neural networks with regularized horseshoe priors and scale mixture priors. AIMS Math 10(9):21929–21952
16. Croce F, Hein M (2020) Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proceedings of the 37th International Conference on Machine Learning (ICML), pp 2206–2216
17. Engelen G, Rimmer V, Joosen W (2021) Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. In: Proceedings of the IEEE Security and Privacy Workshops (SPW), pp 7–12
18. Feng T, Rudner TGJ, Tsilivis N, Kempe J (2024) Attacking Bayes: on the adversarial robustness of Bayesian neural networks. arXiv:2404.19640
19. Gal Y, Ghahramani Z (2016) Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning (ICML), pp 1050–1059
20. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR)
21. Heydari V, Nyarko K (2025) Enhancing adversarial robustness in network intrusion detection: a novel adversarially trained neural network approach. Electronics 14(16):3249. https://doi.org/10.3390/electronics14163249

22. Issa MM, Aljanabi M, Muhialdeen HM (2024) Systematic literature review on intrusion detection systems: research trends, algorithms, methods, datasets, and limitations. J Intell Syst 33(1):20230248. https://doi.org/10.1515/jisys-2023-0248
23. Kingma DP, Welling M (2014) Auto-encoding variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR)
24. Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. Adv Neural Inf Process Syst 30(NeurIPS):6402–6413
25. MacKay DJC (1992) A practical Bayesian framework for backpropagation networks. Neural Comput 4(3):448–472
26. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: Proceedings of the 6th International Conference on Learning Representations (ICLR)
27. McCarthy A (2023) Methods for improving robustness against adversarial machine learning attacks. Dissertation, University of the West of England, Bristol
28. Miyato T, Maeda S-I, Koyama M, Ishii S (2019) Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans Pattern Anal Mach Intell 41(8):1979–1993
29. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of the Military Communications and Information Systems Conference (MilCIS), Canberra, pp 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
30. Neal RM (1996) Bayesian learning for neural networks. Springer, New York
31. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A (2017) Practical black-box attacks against machine learning. In: Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA CCS), pp 506–519
32. Rice L, Wong E, Kolter JZ (2020) Overfitting in adversarially robust deep learning. In: Proceedings of the 37th International Conference on Machine Learning (ICML), pp 8093–8104
33. Rigaki M, Garcia S (2018) Bringing a GAN to a knife-fight: adapting malware communication to avoid detection. In: Proceedings of the IEEE Security and Privacy Workshops (SPW), pp 70–75
34. Sharafaldin I, Habibi Lashkari A, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), pp 108–116
35. Sheatsley R, Hoak B, Pauley E, Beugin Y, Weisman MJ, McDaniel P (2021) On the robustness of domain constraints. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), pp 495–515
36. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR)
37. Tramèr F, Carlini N, Brendel W, Madry A (2020) On adaptive attacks to adversarial example defenses. Adv Neural Inf Process Syst 33(NeurIPS):1633–1645
38. Wicker M, Laurenti L, Patane A, Kwiatkowska M (2021) Probabilistic safety for Bayesian neural networks. In: Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI), pp 1198–1207
39. Wicker M, Laurenti L, Patane A, Kwiatkowska M (2023) Adversarial robustness certification for Bayesian neural networks. arXiv:2306.13614
40. Ye N, Zhu J (2018) Bayesian adversarial learning. In: Advances in Neural Information Processing Systems 31 (NeurIPS), pp 6892–6901
41. Zhang H, Yu Y, Jiao J, Xing EP, El Ghaoui L, Jordan MI (2019) Theoretically principled trade-off between robustness and accuracy. In: Proceedings of the 36th International Conference on Machine Learning (ICML), pp 7472–7482

Figures

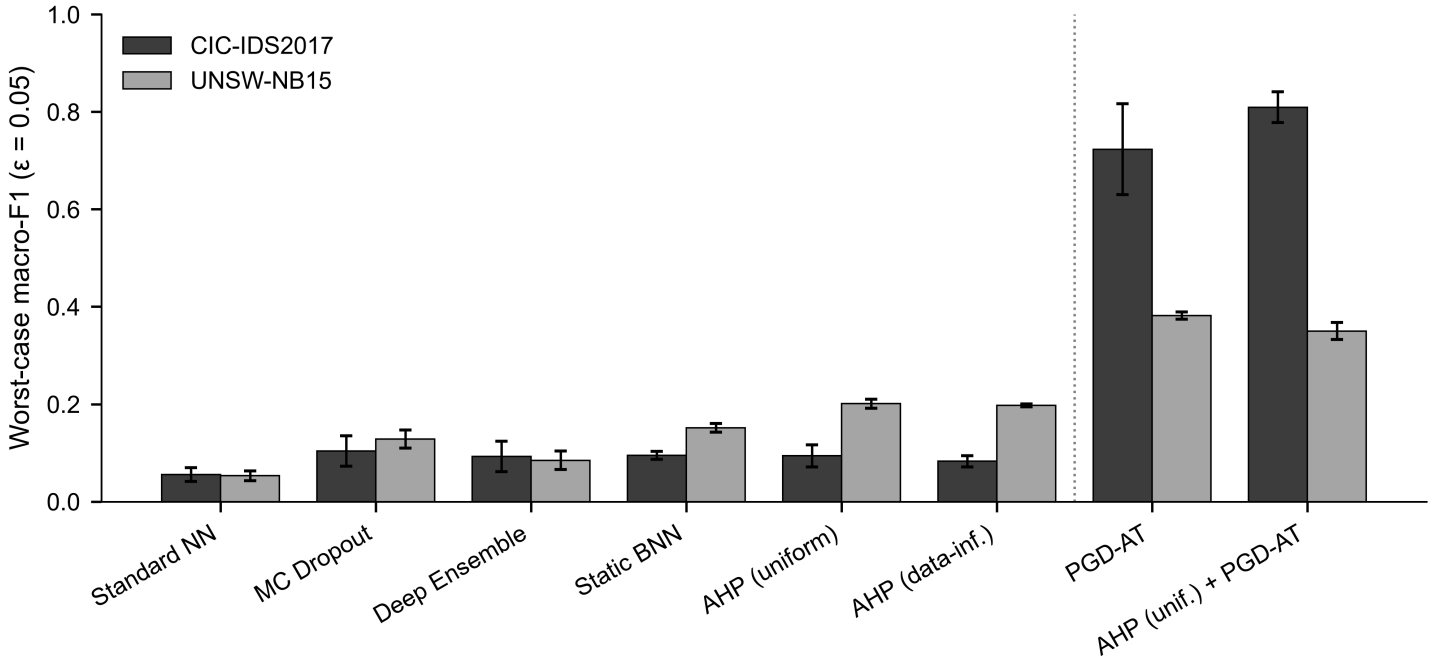


Figure 1

Worst-case macro-F1 at $\varepsilon = 0.05$ on CIC-IDS2017 (dark) and UNSW-NB15 (light): a flow counts as correct only if it is classified correctly on clean data and under every attack (PGD-EOT, APGD-CE, APGD-DLR, Square). Bars show the mean over three seeds, error bars one standard deviation. Models to the right of the dotted line are adversarially trained

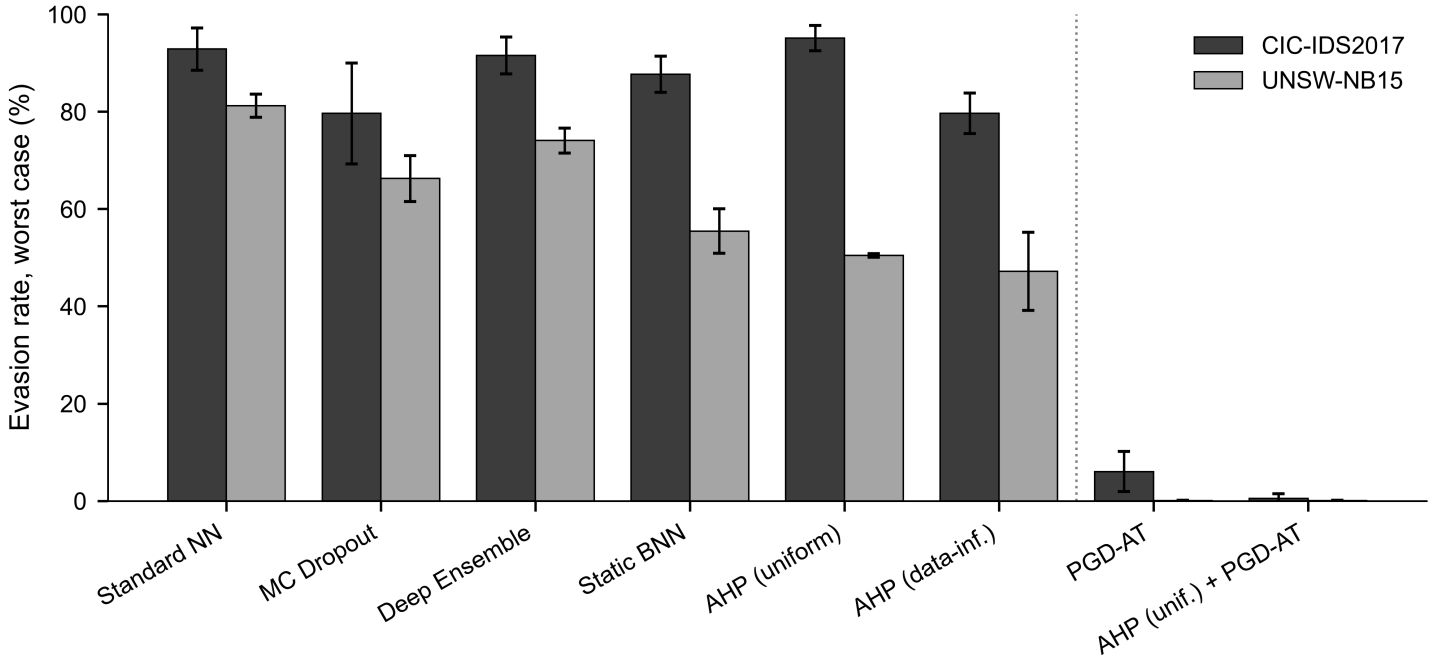


Figure 2

Worst-case evasion rate at $\varepsilon = 0.05$: among malicious flows detected on clean data, the percentage classified as benign by either a targeted PGD-EOT or a targeted square attack. Bars show the mean over three seeds, error bars one standard

deviation. On UNSW-NB15 both adversarially trained models have 0.0% evasion, so their bars are not visible. Models to the right of the dotted line are adversarially trained

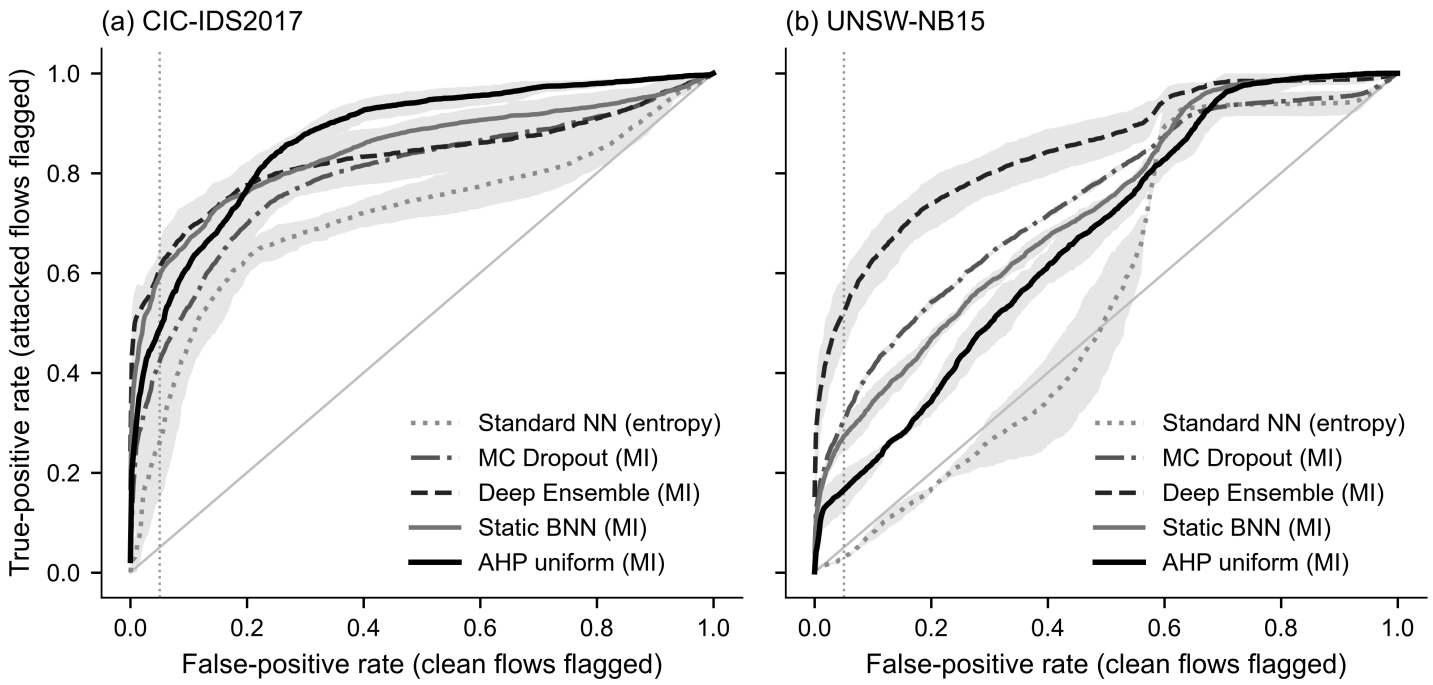


Figure 3

ROC curves for detecting PGD-perturbed flows ($\varepsilon = 0.05$) from their uncertainty on (a) CIC-IDS2017 and (b) UNSW-NB15: predictive entropy for the Standard NN and mutual information (MI) for the other models. Curves are averaged over three seeds on a common false-positive-rate grid; light grey bands show one standard deviation. The dotted vertical line marks the 5% false-positive rate used by the detector