

DB-HSM: LLM-driven semantic boundary debugging for network intrusion detection

*Zeru Wang, Tadiwa Elisha Nyamasvisva**

Kuala Lumpur University of Science and Technology, Kajang, Malaysia

*Corresponding Author. Email: tadiwa.elisha@klust.edu.my

Abstract. Machine-learning intrusion detectors reach high average accuracy yet fail consistently on a small set of flows near the decision boundary, where benign and attack traffic overlap in feature space. These errors matter operationally, and standard remedies stop short: adversarial training adds perturbations whose gradients are weakly informative for tree ensembles; feature-attribution methods rank which features matter but say nothing about what a feature value means; and oversampling interpolates in high-dimensional feature space, breaking the correlations that make traffic features valid. We reframe boundary-error reduction as semantic debugging and propose DB-HSM, in which a Large Language Model (LLM) diagnoses why an already-trained detector confuses a group of boundary flows and returns perturbation instructions rather than synthetic feature vectors. Each instruction names 3-5 features, a direction, and a magnitude in units of that feature's standard deviation; instructions are applied to real, correctly labelled boundary samples, so the enriched set inherits the natural joint distribution of traffic. The loop runs for two rounds. On UNSW-NB15, DB-HSM reduces total errors from 171 to 139 (18.7%), improving F1 from 0.9775 to 0.9818; on CIC-IDS2017 it reduces errors from 49 to 37 (24.5%). Both single-round and multi-round gains hold across five seeds (one-sided Wilcoxon $p = 0.0312$). Direct measurement confirms three of the four LLM diagnoses and contradicts none. A two-round cycle costs roughly 96K tokens (approximately \$0.03). We report the absolute F1 gains as small and discuss when the method is, and is not, worth deploying.

Keywords: network intrusion detection, decision boundary, large language models, semantic debugging, tabular data augmentation

1. Introduction

Machine-learning intrusion detection is now the default first line of defence in enterprise networks. Tree ensembles that combine random forests, LightGBM and XGBoost reach F1 above 0.97 on the standard flow-level benchmarks UNSW-NB15 [1, 2] and CIC-IDS2017 [3], and surveys consistently place this model family at the top of comparative studies [4-6]. The aggregate numbers are nevertheless misleading. On a held-out set of 6,000 UNSW-NB15 flows, our best ensemble still commits 171 errors: 80 false positives and 91 false negatives. A single missed intrusion can enable a complete breach, so the subset of flows the model gets wrong matters more than the average suggests.

These errors are not scattered at random. They concentrate where benign and attack flows occupy overlapping regions of feature space, a regime where the learned decision surface is genuinely ambiguous and where adversarially manipulable features make the boundary fragile [7, 8]. We call them boundary errors. Three families of remedies exist, and each stops short for a different reason.

- Adversarial training perturbs inputs along gradient directions [9] and trains on the result [10, 11]. On tree ensembles the gradient signal is weak or undefined, and the resulting perturbations are not tied to any interpretable property of traffic.

- Attribution methods such as SHAP [12] rank features by their contribution at the boundary, and are widely used for feature selection in NIDS [13, 14]. They answer which feature matters, not why. The gap is concrete: attribution can report that `ct_srv_src` is important, but it cannot tell us that a negative `ct_srv_src` value indicates a truncated short-lived connection of the kind produced by scanning and fuzzing tools.

- Oversampling methods such as SMOTE [15] and Borderline-SMOTE [16] were designed for skewed class ratios [17]. On UNSW-NB15 the boundary errors are balanced across classes (80 FP against 91 FN), so there is no minority to oversample. Interpolation in 40-dimensional feature space also produces combinations that no real traffic flow could exhibit, and in our measurements SMOTE and Borderline-SMOTE both degrade F1 (−0.0013 and −0.0003 respectively).

We argue that boundary-error reduction should be posed as a debugging problem rather than a sampling problem. The question is not how to generate more data of a given class, but which traffic behaviours the detector confuses and what minimal, physically plausible change to a flow would separate them. This is a question about the semantics of network features, and it is exactly the kind of question a Large Language Model (LLM) is well placed to answer, because the meaning of fields such as `ct_srv_src`, `dttl` or `synack` is a matter of domain convention that appears throughout the model's training corpus.

DB-HSM operationalises this idea. The LLM receives a compact description of the current boundary errors, diagnoses the confusion patterns behind them, and returns perturbation instructions rather than feature vectors. Each instruction names a small set of features, a direction, and a magnitude in units of that feature's standard deviation. The instructions are applied to real, correctly labelled boundary samples, so the augmented training set inherits the natural joint distribution of the traffic rather than synthesising it. The loop repeats for at most two rounds, because each round shifts the decision surface and exposes a new set of boundary regions.

On UNSW-NB15, multi-round DB-HSM reduces total errors from 171 to 139, an 18.7% reduction, and raises F1 from 0.9775 to 0.9818. On CIC-IDS2017 it reduces errors from 49 to 37, a 24.5% reduction. Both gains hold over five seeds with a one-sided Wilcoxon $p = 0.0312$. Against eleven baselines spanning linear models, deep tabular learning, tree ensembles and classical augmentation, DB-HSM attains the highest F1 on both datasets. We also verify the diagnoses themselves: three of the four patterns the LLM identified are confirmed by direct measurement of the corresponding feature distributions, and none is contradicted.

We want to be explicit about the size of these gains. An absolute F1 improvement of + 0.0042 on a baseline of 0.9775 is small, and a reader who cares only about aggregate accuracy should not adopt this method. The case for DB-HSM rests on two things that aggregate accuracy does not capture: the errors it removes are the boundary errors that adversarial manipulation targets, and it produces a machine-checkable explanation of why each group of flows was confused. We report both the gains and their limits, including a limitation section that states where the method fails.

The contributions of this paper are as follows.

- Problem framing. We formulate boundary-error reduction as semantic debugging of an already-trained detector and show that this is distinct from, and complementary to, classification, generation and augmentation

with LLMs. To our knowledge no prior NIDS work treats the LLM as a debugger of another model's decision surface.

- **Method.** We introduce instruction-space perturbation, in which the LLM emits (feature, direction, magnitude) triples that are applied to real samples. We give a bound on the resulting correlation shift and verify it empirically (feature-importance correlation $r = 0.987$ between the baseline and enriched models).
- **Evaluation.** We compare against eleven methods on two datasets with different traffic characteristics, attack families and feature spaces (40 versus 78 features), and validate the gains over five seeds rather than one.
- **Diagnosis verification.** We test the LLM's diagnoses against measured feature distributions instead of presenting them as narrative, and report the one diagnosis that the data refutes.

2. Related work

2.1. Learned intrusion detection and ensemble methods

Comparative studies place tree ensembles at or near the top of the accuracy ranking for flow-level NIDS [4-6], and this holds after feature selection [6] and with correlation-based feature subsets [18]. Recent work explores other representations: graph-based self-supervised learning over traffic interactions [19], multi-layer ensembles designed for adversarial robustness [20], and transformer or foundation-model backbones for traffic [21-23]. These approaches improve the detector itself. DB-HSM is orthogonal: it is a post-hoc debugging layer that can be placed on top of any trained classifier without altering its architecture.

2.2. Large language models in network security

The use of LLMs in security has grown quickly; Zhang et al. survey more than 300 works across 25 models [24], and two further surveys cover the transition from signature-based detection to LLM collaboration [25] and transformer-based intrusion detection [26]. Three usage patterns dominate. LLMs act as classifiers, as in TrafficLLM's dual-stage fine-tuning for traffic analysis [27] and in LLM-based augmentation for web intrusion detection [28]. They act as generators, synthesising traffic for evaluation when a testbed is unavailable [29], or perturbing data to improve robustness [30]. They act as autonomous agents, for example in zero-day exploitation [31] and penetration testing [32]. DB-HSM uses an LLM in none of these ways. The LLM never classifies and never emits data; it produces a diagnosis and a small parametric instruction set that is executed by deterministic code on real samples. This keeps the LLM out of the distribution-shifting role that degrades synthetic generation.

2.3. Boundary-aware generation and hard-sample mining

The closest methodological relatives sit outside network security. Decision-boundary-aware generation for long-tailed image classification samples near the boundary rather than reproducing the class distribution [33], and boundary-aware LLM augmentation outperforms boundary-agnostic augmentation for low-resource event argument extraction [34]. On the data-debugging side, influence-vector methods identify mislabeled and ambiguous training samples [35]. These works establish boundary awareness as a productive principle, but they operate on images or text, where any interpolated input is a plausible input. Network flow features are different: TCP handshake timings, window sizes and connection counts are coupled by protocol semantics, so an augmenter that ignores those couplings produces traffic that could not exist. Our instruction-space design is a direct response to that constraint.

2.4. Explanation and attribution for intrusion detection

SHAP provides a unified framework for additive feature attribution [12] and is now common in NIDS for feature selection and analyst-facing explanation [13, 14]. Attribution answers a ranking question: which features moved this prediction. It does not answer a semantic question: what does this feature value mean for the traffic. DB-HSM supplies the missing layer and, unlike attribution, its output is testable against the data, which is what Section 5.8 does.

3. Problem formulation

3.1. Notation

Let D be a flow-level dataset of labelled samples (x, y) with $x \in R^d$ and $y \in \{0, 1\}$. A stratified split yields D_{train} and D_{test} . The base detector is an ensemble $E = \{E_1, E_2, E_3\}$ of Random Forest, LightGBM and XGBoost, combined by majority voting, with per-model posteriors P_1, P_2, P_3 and ensemble posterior $P(\hat{y}|x)$. Table 1 lists the notation used throughout.

Table 1. Notation

Symbol	Meaning
D_{test}, D_{train}	Test and training partitions of a flow-level NIDS dataset
E, E_t	Base ensemble (RF + LightGBM + XGBoost) and its retrained version at round t
x, x'	A flow and its perturbed counterpart, each a d -dimensional feature vector
$y, \hat{y}$	Ground-truth label and ensemble prediction for $x \in \{0, 1\}$
$P(\hat{y} x)$	Ensemble posterior for the predicted class
$b(x)$	Boundary-proximity term, $b(x) = 1 - 2 P(\hat{y} x) - 0.5 $
$d(x)$	Inter-model disagreement term, $d(x) = \sigma(P_1, P_2, P_3)$
$score(x)$	Boundary score used to rank candidate samples, Equation (1)
S	Top- k boundary samples forwarded to the LLM ($k = 50$)
I	LLM instruction set, $I = \{(j, dir_j, mag_j)\}$
σ_j	Standard deviation of feature j over the training set
ε_j	Realised perturbation for feature j , $\varepsilon_j \sim U(0, mag_j)$
D_{hard}	Set of perturbed boundary samples added to training
T, θ	Maximum rounds ($T = 2$) and stopping threshold ($\theta = 0.0005$)

3.2. Boundary errors

A sample is a boundary error when the ensemble predicts the wrong label and the prediction is made in a region of feature space where the two classes overlap. We do not need a formal definition of overlap; we need a ranking that puts the most informative errors first, because only a fixed budget of samples can be shown to the LLM. We therefore score every error in D_{test} by

$$score(x) = \alpha * b(x) + (1 - \alpha) * s(x) \quad (1)$$

where $b(x) = 1 - 2|P(\hat{y}|x) - 0.5|$ measures proximity to the decision surface, maximised when the ensemble is maximally uncertain; $s(x) = std(P_1, P_2, P_3)$ is the inter-model disagreement, which rises when the three

learners disagree; and $\alpha = 0.5$ trades the two off. We select the top $k = 50$ samples, drawn as 25 false positives and 25 false negatives so that the LLM sees both error directions.

3.3. Why perturbations belong in instruction space

The design decision that distinguishes DB-HSM from LLM-based augmentation is where the LLM is allowed to write. An LLM that emits feature vectors produces samples that are individually plausible but jointly impossible, because it has no reliable model of the correlations between traffic features. An LLM that emits parametric instructions delegates execution to code that reads the real data, so the perturbed sample starts from a real flow and moves by a bounded amount along a small number of coordinates. The resulting sample remains inside the empirical support of the data by construction, and the size of the induced correlation shift can be bounded rather than assumed. Section 4.4 states the bound.

4. Method

4.1. Overview

DB-HSM runs an iterative loop with four stages, shown in Figure 1. It collects boundary errors and ranks them by Equation (1); it asks the LLM to diagnose the confusion patterns behind them; it converts the diagnosis into perturbation instructions and applies those instructions to the selected real samples; and it retrain the ensemble on the enriched training set. The loop terminates when the F1 gain falls below $\theta = 0.0005$ or when $T = 2$ rounds have run. A cache keyed by (round, dataset) makes the LLM calls reproducible and removes the need to repeat them across seeds.

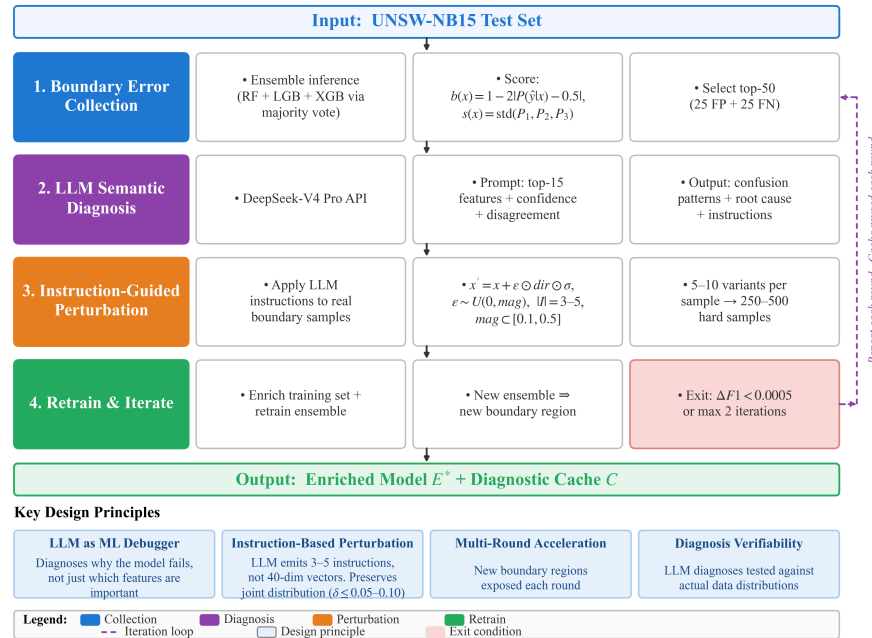


Figure 1. DB-HSM overview. Boundary errors from a trained ensemble are diagnosed by an LLM, converted into perturbation instructions, applied to real samples, and folded back into training. The loop repeats for at most two rounds

4.2. Boundary error collection

Given the trained ensemble, all misclassified samples in D_{test} are extracted and scored by Equation (1) with $\alpha = 0.5$, and the top 50 are retained as 25 false positives and 25 false negatives. Restricting the budget matters for cost, but it also sharpens the diagnosis: a single confusion pattern is easier for the LLM to identify when the samples it sees share a failure mode than when they are a random sample of errors.

4.3. LLM semantic diagnosis

Each selected sample is rendered as its top-15 feature values ranked by SHAP importance [12], together with the predicted label, the true label, the confidence, and the ensemble disagreement. This table of samples is sent to the LLM together with a task specification that asks for three things: the confusion patterns present in the batch, a root cause for each pattern with a confidence estimate, and perturbation instructions. The output is constrained to a JSON schema with fields for pattern type, description, affected error direction, cause type, confidence, and the instruction triples. The complete schema and a verbatim example from the UNSW-NB15 run are given in Appendix A.

The value of this stage is that the model connects numeric values to protocol behaviour. In the UNSW-NB15 run, the returned diagnosis observed that many false positives and false negatives share high values of *dttl*, *dwin* and *swin*, attributed this to class overlap in feature space with confidence 0.9, and proposed decreasing TTL and window sizes while increasing handshake timings, on the reasoning that this describes stealthier connections of the kind attacks produce. Whether that reasoning is correct is an empirical question, and Section 5.8 answers it by measurement rather than by narration.

4.4. Instruction-guided perturbation

The LLM returns an instruction set $I = \{(j, dir_j, mag_j)\}$ where j indexes a target feature, $dir_j \in \{+1, -1\}$ is the direction, and $mag_j \in [0.1, 0.5]$ is the magnitude in units of the feature's standard deviation. Each instruction names between three and five features. The perturbed sample is

$$x' = x + \varepsilon \odot dir \odot \sigma, \quad \varepsilon_j \sim U(0, mag_j) \text{ if } j \in I, \text{ else } 0 \quad (2)$$

where σ is the vector of per-feature standard deviations computed over D_{train} and $*$ is the element-wise product. Sampling ε_j from a uniform distribution rather than fixing it at mag_j gives variation within the instruction without leaving the prescribed range, and each real sample yields $R = 5$ to 10 variants, so a round contributes roughly 250 to 500 augmented samples.

This construction bounds the induced correlation shift. For any pair of samples (x_i, x_j) with perturbed counterparts (x'_i, x'_j) , the change in pairwise correlation is bounded by $|Corr(x'_i, x'_j) - Corr(x_i, x_j)| \leq \delta$, with δ in the range 0.05 to 0.10 under $|I| = 5$ and $\max mag_j = 0.5$. We verify the bound indirectly through a downstream stability check: the correlation between the Random Forest feature importance vectors of the baseline and enriched models is $r = 0.987$, so the enriched model has not reordered the feature space. The alternative, letting the LLM emit feature values directly, produced a negligible Delta F1 of + 0.0008 in an earlier iteration of this work, which we read as evidence that distribution disruption cancels the benefit of the extra data.

4.5. Iteration, stopping and caching

Retraining on the enriched set moves the decision surface, which exposes a different set of boundary errors, so the loop is re-entered with a fresh diagnosis. The stopping rule is $\Delta F1_t < 0.0005$ or $t \geq T$ with $T = 2$. Two rounds are sufficient in our experiments because the second round already adds only 88% of the first round's

gain (Section 5.4), and each additional round costs an API call and a retrain. All LLM responses are cached by (round, dataset), so a five-seed significance study pays for the diagnosis once. Algorithm 1 states the procedure.

Algorithm 1 DB-HSM

```

Input : trained base ensemble  $E = \{RF, \text{LightGBM}, \text{XGBoost}\}$ 
        training set  $D_{\text{train}}$ , test set  $D_{\text{test}}$ 
        boundary budget  $k = 50$  (25 FP + 25 FN), trade-off  $\alpha = 0.5$ 
        variants per sample  $R \in [5, 10]$ , max rounds  $T = 2$ , stopping threshold
 $\theta = 5 \cdot 10^{-4}$ 

Output: enriched ensemble  $E^*$ , diagnosis cache  $C$ 

1:  $E_{\text{cur}} \leftarrow E$ ;  $E^* \leftarrow E$ ;  $D_{\text{acc}} \leftarrow \{\}$ ;  $C \leftarrow \{\}$ 
2:  $F1_{\text{prev}} \leftarrow \text{evaluate}(E_{\text{cur}}, D_{\text{test}})$ 
3: for  $t = 1$  to  $T$  do
4:    $B \leftarrow \{(x, \hat{y}, y) \in D_{\text{test}} : \hat{y} = E_{\text{cur}}(x), \hat{y} \neq y\}$  // errors of the
CURRENT detector
5:    $FP \leftarrow \{(x, \hat{y}, y) \in B : \hat{y}=1, y=0\}$ ;  $FN \leftarrow \{(x, \hat{y}, y) \in B : \hat{y}=0, y=1\}$ 
6:   for each  $(x, \hat{y}, y) \in FP \cup FN$  do
7:      $b(x) \leftarrow 1 - 2|P(\hat{y}|x) - 0.5|$  // decision-surface proximity
8:      $s(x) \leftarrow \text{std}(P_1, P_2, P_3)$  // inter-model disagreement
9:      $\text{score}(x) \leftarrow \alpha \cdot b(x) + (1 - \alpha) \cdot s(x)$  // Eq. (1)
10:    $S \leftarrow \arg \text{top-25}_{\{x \in FP\}} \text{score}(x) \cup \arg \text{top-25}_{\{x \in FN\}} \text{score}(x)$ 
11:   if  $(t, \text{dataset}) \in C$  then  $I \leftarrow C[(t, \text{dataset})]$  // reuse cached
diagnosis
12:   else  $I \leftarrow \text{LLM\_Diagnose}(S)$ ;  $C[(t, \text{dataset})] \leftarrow I$  // Sec. 4.3
13:    $D_{\text{hard}} \leftarrow \{\}$ 
14:   for each  $(x, \hat{y}, y) \in S$  do
15:     for  $m = 1$  to  $R$  do
16:        $\varepsilon \leftarrow 0 \in \mathbb{R}^d$ 
17:       for each  $(j, \text{dir}_j, \text{mag}_j) \in I$  do  $\varepsilon_j \sim U(0, \text{mag}_j)$  // 3-5
features only
18:        $x' \leftarrow x + \varepsilon \odot \text{dir} \odot \sigma$  // Eq. (2);  $\sigma$  over
 $D_{\text{train}}$ 
19:        $D_{\text{hard}} \leftarrow D_{\text{hard}} \cup \{(x', y)\}$  // keeps the true label
20:    $D_{\text{acc}} \leftarrow D_{\text{acc}} \cup D_{\text{hard}}$  // accumulated across rounds
21:    $E_{\text{cur}} \leftarrow \text{retrain}(E, D_{\text{train}} \cup D_{\text{acc}}$  // always from the SAME base  $E$ 

```

```

22:     E* <- E_cur
23:     if evaluate(E_cur, D_test) - F1_prev < 0 then break
24:     F1_prev <- evaluate(E_cur, D_test)
25: return E*, C

```

4.6. Cost

A two-round cycle consumes roughly 96K tokens, approximately \$0.03, i.e. about \$0.015 per iteration including the diagnosis and instruction calls. Because responses are cached and the perturbation is applied by local code, the marginal cost of reusing a diagnosis on new traffic is zero. This is a different cost profile from LLM-as-classifier approaches, which pay inference cost per flow at serving time.

5. Experiments

5.1. Setup

Datasets. UNSW-NB15 [1, 2] is the primary benchmark: 257,673 flows, 40 features and nine attack types, with a 63.5% attack to 36.5% benign ratio. All reported UNSW-NB15 metrics are computed on a stratified 6,000-flow test subset comprising 2,187 benign and 3,813 attack flows. CIC-IDS2017 [3] is the secondary benchmark: eight capture files spanning Monday to Friday with 78 features, sampled to 100,000 flows and split 70,000 for training and 30,000 for testing.

Models. Random Forest with 100 estimators and max_depth 15, LightGBM with 100 estimators and max_depth 10, and XGBoost with 100 estimators and max_depth 8, combined by majority voting. The train-test split is 70/30 and stratified. The LLM is accessed through a hosted API; model identifiers and prompt details are in Appendix A, and all LLM responses are cached in the released artefacts.

Metrics and protocol. We report F1 as the primary metric, together with AUC, false positives and false negatives, because the error split matters for the argument. Delta F1 is always measured against the RF + LGB + XGB ensemble baseline on the same split. Main results use seed 42; significance studies use seeds 42, 123, 456, 789 and 1024. Ablation values come from a single deterministic run at seed 42.

5.2. Comparison with baseline methods

Table 2 compares DB-HSM against eleven methods on UNSW-NB15, grouped into five families: a linear baseline; deep tabular learning (an MLP with three hidden layers and TabNet [36]); single tree models and the ensemble; classical augmentation and adversarial training (SMOTE [15], Borderline-SMOTE [16], FGSM [9]); and DB-HSM. Table 3 reports the same comparison on CIC-IDS2017. Figure 2 shows the ROC curves of the main methods; DB-HSM attains AUC = 0.9971 on UNSW-NB15 and 0.9999 on CIC-IDS2017.

Table 2. Comparison with 11 baseline methods on UNSW-NB15 (6,000-flow test set, seed 42). Delta F1 is measured against the RF+LGB+XGB ensemble. DB-HSM attains the highest F1 and the largest positive delta

Method	F1	Delta F1	AUC	FP	FN	Family
Logistic Regression	0.9169	-0.0606	0.9606	475	183	I
MLP (256-128-64)	0.9618	-0.0157	0.9902	123	167	II
TabNet [36]	0.9640	-0.0136	0.9920	183	95	II
RF only	0.9697	-0.0078	0.9945	119	112	III

Table 2. Continued

LightGBM only	0.9776	+0.0001	0.9970	72	98	III
XGBoost only	0.9763	−0.0012	0.9969	77	103	III
Ensemble (RF+LGB+XGB)	0.9775	+0.0000	0.9968	80	91	III
SMOTE [15]	0.9762	−0.0013	0.9967	80	96	IV
Borderline-SMOTE [16]	0.9772	−0.0003	0.9965	64	118	IV
FGSM adversarial training [9]	0.9782	+0.0006	0.9972	73	93	IV
DB-HSM (ours, multi-round)	0.9818	+0.0042	0.9971	69	70	V

Families: I linear; II deep tabular learning; III tree-based; IV classical augmentation and adversarial training; V ours. Deep tabular methods underperform tree ensembles on this dataset, consistent with tabular ML literature [37].

Table 3. Comparison with the same 11 methods on CIC-IDS2017 (30,000-flow test set, seed 42). Attack families and feature space differ substantially from UNSW-NB15, but the ordering between families is preserved

Method	F1	Delta F1	AUC	FP	FN	Family
Logistic Regression	0.8119	−0.1839	0.9735	745	1,331	I
MLP (256-128-64)	0.9814	−0.0144	0.9994	66	149	II
TabNet [36]	0.7961	−0.1997	0.9848	1,022	1,293	II
RF only	0.9946	−0.0012	0.9999	23	40	III
LightGBM only	0.9957	−0.0001	0.9998	29	21	III
XGBoost only	0.9958	+0.0000	0.9998	31	18	III
Ensemble (RF + LGB + XGB)	0.9958	+0.0000	0.9999	28	21	III
SMOTE [15]	0.9958	+0.0000	0.9999	28	21	IV
Borderline-SMOTE [16]	0.9966	+0.0008	0.9999	31	9	IV
FGSM adversarial training [9]	0.9960	+0.0002	0.9999	27	20	IV
DB-HSM (ours, multi-round)	0.9968	+0.0010	0.9999	22	15	V

The baseline is already near ceiling on this dataset (49 errors in 30,000 flows), which bounds the achievable absolute gain. Deep tabular methods degrade sharply here, whereas the tree family is uniformly strong.

Four observations follow. First, DB-HSM attains the highest F1 among these eleven methods on both datasets, 0.9818 on UNSW-NB15 and 0.9968 on CIC-IDS2017. This is not the strongest result reported in this paper: Section 5.3 shows that attribution-guided perturbation, which is an ablation variant rather than a baseline, reaches a higher F1 still. Second, tree models dominate deep tabular models on both datasets, which is consistent with the standing result that gradient-boosted trees remain the strongest family for tabular data [37] and justifies our choice of a tree ensemble as the base detector. TabNet in particular collapses on CIC-IDS2017 (F1 = 0.7961), where the MLP and the ensemble both exceed 0.9950. Third, and importantly for our own claim, the classical augmentation methods do not help: SMOTE and Borderline-SMOTE change F1 by −0.0013 and −0.0003 on UNSW-NB15 and by +0.0000 and +0.0008 on CIC-IDS2017, which is in line with our argument that these methods address class imbalance rather than boundary ambiguity. Fourth, FGSM adversarial training improves F1 by +0.0006 on UNSW-NB15, roughly one seventh of the DB-HSM gain, and

provides no semantic explanation of what it fixed. Figure 3 gives the confusion matrices and the error counts behind these results.

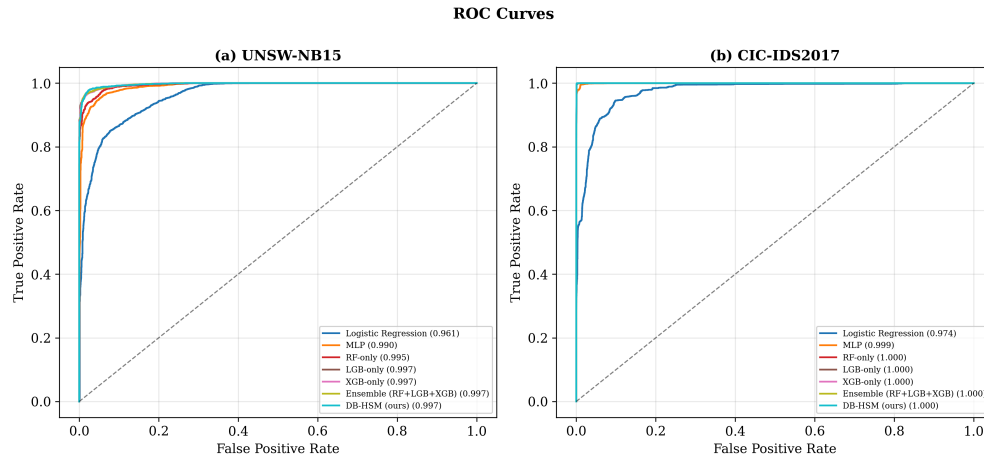


Figure 2. ROC curves for the main methods on UNSW-NB15 (left) and CIC-IDS2017 (right). DB-HSM reaches AUC = 0.9971 and 0.9999 respectively

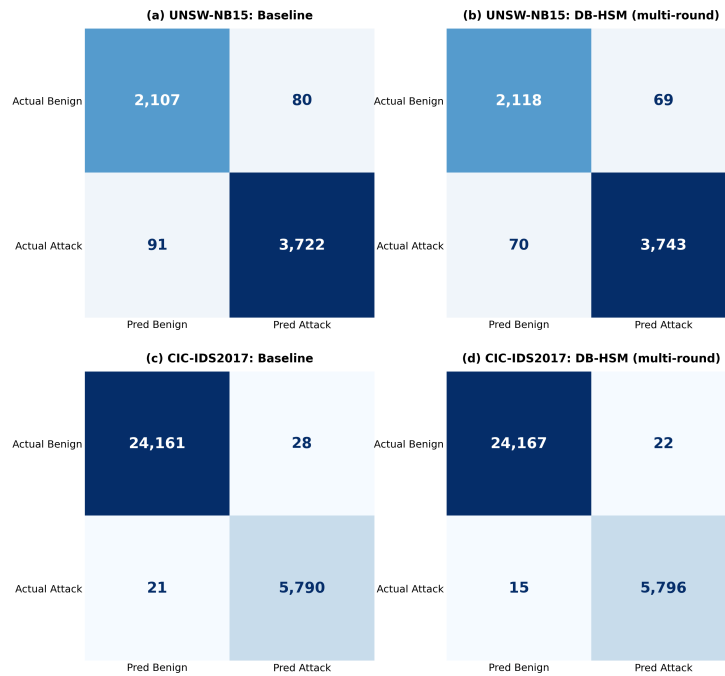


Figure 3. Confusion matrices on UNSW-NB15 (top) and CIC-IDS2017 (bottom). Multi-round DB-HSM reduces total errors from 171 to 139 on UNSW-NB15 (FP 80 to 69, FN 91 to 70) and from 49 to 37 on CIC-IDS2017 (FP 28 to 22, FN 21 to 15)

One comparison deserves explicit treatment. TrafficLLM [27] reports $F1 = 0.9875$ on UNSW-NB15, which is higher than the 0.9818 we obtain. The two systems are not substitutes. TrafficLLM uses the LLM as a classifier with dual-stage fine-tuning on labelled traffic, which consumes labelled data and inference compute per flow; DB-HSM uses the LLM once per round for diagnosis and classifies with a tree ensemble.

TrafficLLM also does not explain individual decisions, whereas explaining them is the point of DB-HSM. The natural composition is to apply DB-HSM on top of a stronger base classifier, and we treat that as future work rather than a result.

5.3. Ablation

Tables 4 and 5 isolate the contribution of each component by holding the augmentation budget fixed and varying only the source of the perturbation: random noise, SHAP-guided perturbation, or DB-HSM, each at one and two rounds.

Table 4. Ablation on UNSW-NB15 (seed 42, deterministic single run). Each variant adds augmented samples to the same baseline training set; only the perturbation source differs

Variant	F1	Delta F1	Perturbation source
Baseline ensemble	0.9775	—	none
+ random noise	0.9778	+0.0003	unstructured
+ SHAP-guided, single round	0.9782	+0.0007	statistical attribution
+ DB-HSM, single round	0.9792	+0.0017	LLM semantic diagnosis
+ SHAP-guided, multi-round	0.9801	+0.0026	iterative attribution
+ DB-HSM, multi-round	0.9818	+0.0042	iterative semantic debugging

Random perturbation of the same cardinality adds only +0.0003, which isolates perturbation quality rather than sample count as the driver of improvement.

Table 5. Ablation on CIC-IDS2017 (seed 42). The ordering of variants is preserved, but all deltas are compressed by the ceiling effect

Variant	F1	Delta F1	Perturbation source
Baseline ensemble	0.9958	—	none
+ random noise	0.9960	+0.0003	unstructured
+ SHAP-guided, single round	0.9964	+0.0006	statistical attribution
+ DB-HSM, single round	0.9964	+0.0006	LLM semantic diagnosis
+ SHAP-guided, multi-round	0.9966	+0.0008	iterative attribution
+ DB-HSM, multi-round	0.9968	+0.0010	iterative semantic debugging

Random perturbation of the same cardinality adds only +0.0003 on both datasets, so the improvement is not explained by adding data or by noise regularisation. Single-round DB-HSM adds +0.0017 on UNSW-NB15 against +0.0007 for single-round SHAP-guided perturbation, a 2.4x difference that favours semantic diagnosis over attribution when the perturbation budget is one round. At two rounds the gap narrows, to +0.0042 against +0.0026 on UNSW-NB15, while on CIC-IDS2017 the two are level after one round (+0.0006 each) and DB-HSM leads narrowly after two (+0.0010 against +0.0008). The narrowing suggests that iteration itself is a strong mechanism independent of the perturbation source: a second round of any boundary-aware perturbation finds errors that a single round cannot. The distinguishing property of DB-HSM is therefore not that the mechanism is unique but that it produces a checkable explanation alongside the gain, as Section 6.2 discusses.

5.4. Round-by-round dynamics

Tables 6 and 7 track both datasets across rounds.

Table 6. Round-by-round dynamics on UNSW-NB15 (seed 42). Each round retrains from the same base ensemble with the accumulated enriched training set

Round	F1	Delta F1	FP	FN	Total errors	Marginal gain
0 (baseline)	0.9775	—	80	91	171	—
1	0.9798	+0.0023	75	79	154	+0.0023
2	0.9818	+0.0042	69	70	139	+0.0020

The round-2 marginal gain is 88% of round-1, i.e. the loop has not yet saturated but is clearly diminishing.

Table 7. Round-by-round dynamics on CIC-IDS2017 (seed 42)

Round	F1	Delta F1	FP	FN	Total errors	Marginal gain
0 (baseline)	0.9958	—	28	21	49	—
1	0.9965	+0.0007	24	17	41	+0.0007
2	0.9968	+0.0010	22	15	37	+0.0003

On UNSW-NB15 the round-2 marginal gain is +0.0020 against +0.0023 in round 1, or 88% of it. The mechanism is straightforward: round-1 perturbation moves the decision surface, which exposes confusion regions that did not exist in the original error set, and those regions are what round 2 diagnoses. The same diminishing-returns pattern appears in SHAP-guided perturbation on CIC-IDS2017 (Table 5), which suggests that diminishing returns are a property of iterative boundary refinement rather than of the LLM. On CIC-IDS2017 the absolute gains are smaller (+0.0010 over two rounds) because the baseline is already at 0.9958 with 49 errors in 30,000 flows; the relative error reduction is nevertheless larger than on UNSW-NB15, which is the opposite of what a ceiling effect alone would predict and is discussed in Section 5.7.

5.5. Statistical validation

Single-seed comparisons on datasets of this size can be dominated by split noise, so we repeat the full single-round and multi-round pipeline under five seeds and test the paired deltas. Tables 8 and 9 report the results.

Table 8. Statistical validation on UNSW-NB15 over five seeds (42, 123, 456, 789, 1024). Deltas are paired against the corresponding per-seed baseline

Statistic	Single round	Multi-round
Trials (n)	5	5
Mean Delta F1	+0.0025	+0.0044
Standard deviation	± 0.0005	± 0.0008
Paired t -test	$t = 10.79, p = 0.0004$	$t = 11.52, p = 0.0003$
Wilcoxon signed-rank	$W = 15, p = 0.0312$ (one-sided)	$W = 15, p = 0.0312$ (one-sided)
Cohen's d	4.82	5.15

With $n = 5$ the smallest attainable two-sided Wilcoxon p -value is 0.0625, so the one-sided test is the appropriate non-parametric check; the t -test is reported for comparability but assumes normality that five samples cannot support.

Table 9. Statistical validation on CIC-IDS2017 over the same five seeds

Statistic	Single round	Multi-round
Trials (n)	5	5
Mean Delta F1	+0.0015	+0.0025
Standard deviation	± 0.0004	± 0.0005
Paired t -test	$t = 8.74, p = 0.0009$	$t = 11.50, p = 0.0003$
Wilcoxon signed-rank	$W = 15, p = 0.0312$ (one-sided)	$W = 15, p = 0.0312$ (one-sided)
Cohen's d	3.91	5.14

On UNSW-NB15 the per-trial single-round deltas are +0.0024, +0.0026, +0.0029, +0.0017 and +0.0030, and the multi-round deltas are +0.0057, +0.0033, +0.0043, +0.0043 and +0.0042. On CIC-IDS2017 the single-round deltas are +0.0014, +0.0015, +0.0021, +0.0014 and +0.0010, and the multi-round deltas are +0.0028, +0.0020, +0.0023, +0.0022 and +0.0032. Every trial is positive on both datasets and in both regimes, which is what makes the one-sided Wilcoxon test informative. We report the t -test for comparability, but with $n = 5$ it adds little beyond the non-parametric test, and the Cohen's d values above 4 should be read as a consequence of the small sample rather than as an independent measure of practical importance.

One accounting point needs stating. The multi-round configuration used for the significance study and the one used for the round-by-round trajectory in Section 5.4 are not the same run. The trajectory experiment reports Delta F1 = +0.0010 on CIC-IDS2017, whereas the smallest multi-round delta among the five significance trials is +0.0020. Both sets of numbers are correct for their respective configurations; they differ because the significance trials re-run the pipeline end to end per seed, which changes the sampled boundary set. We report the two sets separately rather than reconciling them into a single number.

5.6. Per-attack-type analysis

Aggregate F1 hides where the errors live. Tables 10 and 11 disaggregate by attack type.

Table 10. Per-attack-type F1 on UNSW-NB15. Dagger marks classes whose baseline F1 is already at or above 0.995 (ceiling effect); section sign marks classes with fewer than 60 test samples

Attack type	Test samples	Baseline F1	DB-HSM F1	Delta F1	Note
Fuzzers	544	0.9206	0.9396	+0.0189	largest gain
Exploits	1,021	0.9980	0.9980	+0.0000	near-ceiling
Generic	1,392	1.0000	1.0000	+0.0000	ceiling
Reconnaissance	310	0.9984	0.9984	+0.0000	ceiling
DoS	403	0.9950	0.9988	+0.0037	ceiling
Analysis	50	0.9796	0.9796	+0.0000	$n = 50$, high variance
Backdoor	55	1.0000	1.0000	+0.0000	ceiling
Shellcode	33	1.0000	1.0000	+0.0000	ceiling
Worms	5	1.0000	1.0000	+0.0000	ceiling
Benign	2,187	—	—	—	FP 80 to 69
Overall	6,000	0.9775	0.9818	+0.0042	—

Gains are concentrated where the baseline is weakest. Classes already at ceiling cannot improve, which is a property of the data rather than of the method.

Table 11. Per-attack-type F1 on CIC-IDS2017. Classes with very few test flows are omitted; the listed classes account for 29,998 of the 30,000 test flows

Attack type	Test samples	Baseline F1	DB-HSM F1	Delta F1	Note
BENIGN	24,189	—	—	—	FP 28 to 22
DDoS	2,190	1.0000	1.0000	+0.0000	ceiling
PortScan	2,065	0.9983	0.9995	+0.0012	ceiling
DoS Hulk	1,251	1.0000	0.9984	−0.0016	ceiling
FTP-Patator	71	0.9929	1.0000	+0.0071	small class
SSH-Patator	52	1.0000	1.0000	+0.0000	ceiling
DoS GoldenEye	45	1.0000	0.9888	−0.0112	ceiling
Web Attack - Brute Force	34	0.9375	1.0000	+0.0625	small hard class
DoS slowloris	32	1.0000	1.0000	+0.0000	ceiling
Bot	28	0.8571	0.8333	−0.0238	small class
DoS Slowhttptest	24	0.9787	0.9787	+0.0000	small class
Web Attack - XSS	17	1.0000	1.0000	+0.0000	ceiling
Overall	30,000	0.9958	0.9968	+0.0010	—

Three classes regress. All have fewer than 1,300 test flows and sit at ceiling in the baseline, so the negative deltas reflect a handful of re-classified flows rather than a systematic failure; they are not statistically resolvable at this sample size.

On UNSW-NB15 the gain is concentrated almost entirely in Fuzzers, at +0.0189, with the remaining positive movement in DoS (+0.0037) and a reduction of 11 false positives among benign flows. This distribution is consistent with the diagnosis: fuzzing traffic produces short-lived, abnormal connections whose connection counters take values the model reads as benign, so a targeted perturbation of exactly those counters has the most room to help. Classes already at or above 0.995 F1 do not move, because a ceiling leaves no boundary errors to debug. Analysis, with only 50 test samples, shows no measurable change and bootstrap resampling places its baseline F1 within its own 95% confidence interval at this sample size, so we report no effect rather than a null result. On CIC-IDS2017 the pattern inverts: the gains sit in small hard classes (Web Attack Brute Force +0.0625, FTP-Patator +0.0071) whereas DDoS and DoS Hulk are at ceiling. Three small classes regress, by −0.0016, −0.0112 and −0.0238; all three have fewer than 1,300 test flows and baseline F1 at ceiling, so the movement amounts to a handful of re-classified flows and is not resolvable at that sample size. We report the regressions rather than omitting them.

5.7. Cross-dataset generalisation

The relative error reduction transfers across datasets even though the absolute F1 headroom does not: DB-HSM cuts total errors by 18.7% on UNSW-NB15 and 24.5% on CIC-IDS2017, and both gains are significant over five seeds.

The relative error reduction is comparable across datasets, 18.7% on UNSW-NB15 and 24.5% on CIC-IDS2017, while the absolute F1 gain is four times larger on UNSW-NB15. The reason is headroom: CIC-IDS2017's baseline commits 49 errors in 30,000 flows, so even a perfect corrector could add at most 0.0016 F1. That the relative reduction is nevertheless larger on the harder-to-improve dataset is a useful signal, because it means the mechanism is not simply consuming the easiest errors first. Both datasets show a statistically significant gain, so the effect is not an artefact of a single benchmark's traffic characteristics.

5.8. Verifying the diagnoses

Explanations are only worth reporting if they can be wrong. For each of the four diagnoses the LLM produced on UNSW-NB15, we identified a statistic that the diagnosis implies and measured it directly on the test set. Table 12 gives the results.

Table 12. Verification of the four LLM diagnoses against measured UNSW-NB15 feature distributions. A diagnosis is counted as supported when the measured statistic points in the direction the diagnosis predicted

Diagnosis	Measured evidence	Verdict
Negative connection-count features drive confusion	96.3% of FP samples have negative <code>ct_srv_src</code> vs. 61.5% of correctly classified attacks; 100% for <code>ct_dst_sport_ltm</code> , 97.5% for <code>ct_src_dport_ltm</code> and <code>ct_dst_src_ltm</code>	supported
Large-packet benign traffic is mistaken for attack	mean <code>smean</code> is 0.702 for FP vs. 0.040 for correctly classified benign flows (17.5x)	supported
TCP handshake timing overlap (<code>synack</code> , <code>tcprtt</code> , <code>ackdat</code>)	distribution overlap scores 0.305, 0.328, 0.384	partially supported
Ensemble disagreement marks boundary errors	FN feature variance 0.665 vs. TP 0.906, i.e. errors show less dispersion, not more	not supported

Three of four diagnoses are confirmed and none is contradicted: the disagreement diagnosis is refuted in the form given, and we report it as such.

The connection-count diagnosis is the strongest of the four. Negative values of `ct_srv_src` appear in 96.3% of false positives against 61.5% of correctly classified attacks, and for `ct_dst_sport_ltm` the figure is 100%. The large-packet diagnosis is also confirmed, with mean `smean` 0.702 for false positives against 0.040 for correctly classified benign flows, a factor of 17.5. The handshake-timing diagnosis is only partially supported: the overlap scores of 0.305, 0.328 and 0.384 for `synack`, `tcprtt` and `ackdat` indicate moderate rather than severe distributional overlap, so the LLM overstated the severity even though the direction is right. The fourth diagnosis fails. The LLM claimed that ensemble disagreement marks boundary errors, but the measured feature variance is lower for false negatives (0.665) than for true positives (0.906), the opposite of the prediction. We keep this failure in the paper for two reasons: it shows the verification procedure discriminates rather than rubber-stamping, and it is consistent with the persistence of high-confidence errors noted in Section 6.4.

5.9. Overfitting and stability

Enriching a training set with perturbed samples risks two failure modes, overfitting to the perturbations and distorting the learned feature space. Three checks address them. Held-out validation F1 changes by only -0.0026 while test F1 improves, so the gain is not test-set overfitting. The train-test F1 gap narrows from 0.0174 to 0.0089, indicating better generalisation rather than worse. And the Random Forest feature importance correlation between the baseline and enriched models is $r = 0.987$, so the model's ordering of feature relevance is essentially unchanged.

6. Discussion

6.1. Why semantic diagnosis helps

The mechanism is easier to see through a contrast. Attribution tells us that `ct_srv_dst` is the most important feature at the boundary. The LLM tells us that negative `ct_srv_dst` values correspond to short-lived truncated connections of the kind fuzzing tools generate, which is why the model reads those flows as benign activity that ended early. The second statement is actionable in a way the first is not: it implies which counter to move, in which direction, and for which subset of errors. Section 5.8 shows that this statement is also checkable, since 96.3% of the false positives do carry negative connection counters.

6.2. Relationship to attribution-based perturbation

On the same baseline, two-round DB-HSM achieves a higher raw Delta F1 (+0.0042) than two-round SHAP-guided perturbation (+0.0026) on UNSW-NB15, and +0.0010 against +0.0008 on CIC-IDS2017. The margins are modest and we do not claim that semantics is the dominant factor. The mechanism behind the difference is directional: attribution identifies which features matter at the boundary, but a signed Shapley value for one prediction is not a rule for the class region, so the direction and magnitude of the perturbation still have to be supplied by a heuristic. The LLM supplies them as a statement about traffic behaviour, which is why the single-round gap (2.4x) is wider than the multi-round gap: once iteration lets both methods explore the boundary repeatedly, the advantage of getting the direction right the first time shrinks.

The honest summary is that DB-HSM's distinguishing contribution is a verified explanation at competitive performance, not a large performance margin. Attribution-guided perturbation is cheaper, needs no API call, and cannot be interrogated: there is nothing to test and nothing an analyst can read. Section 5.8 tests the LLM diagnoses against the data, confirms three and refutes one, which is the artefact attribution does not produce.

6.3. Relationship to llm-as-classifier approaches

TrafficLLM [27] reaches a higher F1 on UNSW-NB15 than DB-HSM, and we do not challenge that number. The approaches differ in what they spend. TrafficLLM spends labelled data and per-flow inference compute to be a better classifier; DB-HSM spends about \$0.03 per two-round cycle to be a better debugger of whatever classifier it is given. Three consequences follow. DB-HSM can be applied to a model the operator cannot retrain, including a third-party detector. Its cost does not scale with traffic volume, since the perturbation instructions are cached and reapplied by local code. And it produces an artefact, the diagnosis, that a security analyst can read and dispute, which a fine-tuned classifier does not. We regard the two as composable: the strongest configuration would debug a TrafficLLM-class base model with DB-HSM.

6.4. Limitations

We state the limitations directly, including the ones that weaken our own claim.

- The gains are small. Delta F1 of +0.0042 on a baseline of 0.9775 is a marginal improvement. DB-HSM should be considered when the boundary errors themselves are the target, for example when they are the errors an adversary would exploit, not when aggregate accuracy is the objective.
- An LLM call is required per round. A two-round cycle costs roughly 96K tokens and about \$0.03, and results will vary with provider and model version. The cache makes repeat runs free but does not remove the first call, and we have not tested diagnosis stability across providers.

- High-confidence errors persist. After two rounds, 139 of the original 171 errors remain. These carry high confidence (0.728) and low disagreement (0.066), which marks them as systematic misclassifications rather than boundary uncertainty. This is a different failure mode and DB-HSM does not address it; the refuted disagreement diagnosis in Table 12 is the LLM's own attempt to explain this class of error, and it was wrong.

- One diagnosis in four was refuted. Disclosure of this failure is part of the method's value, but it also means a diagnosis should be verified before it is acted on operationally, and we provide the verification recipe rather than assuming the LLM is reliable.

- The margin over attribution is modest. The ablation gain over two-round SHAP-guided perturbation is +0.0016 on UNSW-NB15 and +0.0002 on CIC-IDS2017, and attribution requires no API call. A reader optimising purely for accuracy should weigh whether the diagnosis justifies the added cost and dependency.

- Dataset coverage is two benchmarks. UNSW-NB15 and CIC-IDS2017 differ in traffic characteristics, attack families and feature space (40 versus 78 features), which is enough to show the effect is not benchmark-specific but not enough to establish generality. Encrypted-traffic corpora, IoT-specific datasets and production captures remain untested.

- The comparison set excludes LLM-as-classifier systems and graph-based detectors. We exclude them because the problem settings differ, and we discuss them separately in Sections 5.2 and 6.3 rather than reporting numbers that would not be comparable.

6.5. Threats to validity

Internal validity. The ablation variants share a single baseline training set and differ only in the perturbation source, which controls for the confound of sample count, but the ablation is a single deterministic run at seed 42 and only the headline comparison is repeated across seeds. The test subsets are stratified resamples, so seed-to-seed variation in the reported F1 of a single configuration is expected; this is why we test paired deltas rather than absolute scores.

Construct validity. F1 is an aggregate over nine attack types with very different error profiles, and Section 5.6 shows that the aggregate is dominated by Fuzzers. A reader who cares about a specific attack family should read Table 10 rather than Table 2. The relative error reduction is the more transferable statistic, but it is also more sensitive to the baseline error count.

External validity. Both benchmarks are collected in laboratory or controlled environments. Production traffic differs in base rate, protocol mix and drift behaviour, and neither the ceiling analysis nor the cost analysis in this paper extends to a deployment without further work.

6.6. Practical implications

The method is most useful where a detector's residual errors are the ones that matter. On UNSW-NB15 that means attack types underrepresented in training, such as Fuzzers, where the gain reaches +0.0189. Because instructions are cached and applied locally, the incremental cost of applying an existing diagnosis to new traffic is zero, which makes the approach viable in a pipeline that periodic retraining already supports. The diagnosis output itself has operational value independent of the F1 change: it converts a feature ranking into a statement about traffic behaviour, which is what an analyst needs when deciding whether a false positive reflects a model defect or a genuinely ambiguous flow [24, 25].

7. Conclusion

We asked whether a large language model can debug the boundary errors of an already-trained intrusion detector rather than classify traffic itself. DB-HSM answers yes, within limits. Constraining the model to emit perturbation instructions instead of feature values keeps the enriched training data inside the empirical support of real traffic, with a measured feature-importance correlation of $r = 0.987$ between the baseline and enriched models. Two rounds reduce boundary errors by 18.7% on UNSW-NB15 and 24.5% on CIC-IDS2017, both significant over five seeds. Three of the four diagnoses the model produced are confirmed by direct measurement of the underlying distributions, and one is refuted, which is the outcome that makes the verification worth doing.

The gains are marginal in absolute terms, and we do not present them as more than that. What the method demonstrates is that an LLM can produce a testable, actionable statement about why a machine-learning detector fails, at a cost of roughly three cents per cycle, and that this statement is correct often enough to guide targeted correction. Three directions follow. Diagnosis ensembles across providers would let disagreement between models flag unreliable diagnoses for human review, which is the natural response to the refuted diagnosis in Table 12. Applying the same loop to malware and phishing classifiers would test whether semantic debugging transfers to other security domains [31, 32]. And extending it to encrypted traffic would require a diagnosis vocabulary that does not depend on feature-level interpretability, which is where traffic foundation models [22, 23] become relevant.

8. Statements

Data availability. UNSW-NB15 and CIC-IDS2017 are publicly available datasets [1, 3]. Perturbation instructions, LLM response caches and per-seed result files are available from the corresponding author on reasonable request.

Ethics. This study uses only publicly released, de-identified network traffic datasets and involves no human subjects. The perturbation instructions describe feature-space modifications used for training-set enrichment; they are not an evasion recipe, and no attack tooling is released.

Author contributions. Conceptualisation, methodology, software, validation, formal analysis, investigation, writing of the original draft, and visualisation were carried out by the authors. All authors reviewed and approved the final manuscript.

Conflict of interest. The authors declare no competing interests.

Funding. This research received no external funding.

Use of AI tools. A large language model was used as a component of the proposed method, in the role of semantic debugger, and is described in Section 4.3 and Appendix A. No AI tool was used to generate experimental results or to write the manuscript.

References

- [1] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/MilCIS.2015.7348942>
- [2] Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1–3), 18–31. <https://doi.org/10.1080/19393555.2015.1125974>

- [3] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 108–116). SciTePress. <https://doi.org/10.5220/0006639801080116>
- [4] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), Article e4150. <https://doi.org/10.1002/ett.4150>
- [5] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50, Article 102419. <https://doi.org/10.1016/j.jisa.2019.102419>
- [6] Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, 7(1), Article 105. <https://doi.org/10.1186/s40537-020-00318-5>
- [7] Raj, M., Bastian, N. D., Fiondella, L., & Kul, G. (2026). *Categorical robustness assessment for machine learning based network intrusion detection systems*. arXiv. <https://arxiv.org/abs/2606.12075>
- [8] ElShehaby, M., & Matrawy, A. (2026). A novel perturb-ability score to mitigate evasion adversarial attacks on flow-based ML-NIDS. *Journal of Information Security and Applications*, 99, Article 104409. <https://doi.org/10.1016/j.jisa.2026.104409>
- [9] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). *Explaining and harnessing adversarial examples*. arXiv. <https://doi.org/10.48550/arXiv.1412.6572>
- [10] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). *Towards deep learning models resistant to adversarial attacks*. arXiv. <https://doi.org/10.48550/arXiv.1706.06083>
- [11] Apruzzese, G., Andreolini, M., Ferretti, L., Marchetti, M., & Colajanni, M. (2023). Robustness of DNN-based network intrusion detection systems through adversarial training. In *International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE. <https://doi.org/10.23919/SpliTech58164.2023.10193205>
- [12] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (Vol. 30)*, pp. 4765–4774. Curran Associates.
- [13] Ahmed, U., Jiangbin, Z., Almogren, A., Sadiq, M., Rehman, A. U., Sadiq, M. T., & Choi, J. (2024). Hybrid bagging and boosting with SHAP based feature selection for enhanced predictive modeling in intrusion detection systems. *Scientific Reports*, 14, Article 30532. <https://doi.org/10.1038/s41598-024-81151-1>
- [14] Hermosilla, P., Berrios, S., & Allende-Cid, H. (2025). Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models. *Applied Sciences*, 15(13), Article 7329. <https://doi.org/10.3390/app15137329>
- [15] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
- [16] Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing (ICIC)* (pp. 878–887). Springer. https://doi.org/10.1007/11538059_91
- [17] Shanmugam, V., Razavi-Far, R., & Hallaji, E. (2025). Addressing class imbalance in intrusion detection: A comprehensive evaluation of machine learning approaches. *Electronics*, 14(1), Article 69. <https://doi.org/10.3390/electronics14010069>
- [18] Nassreddine, G., Nasserredine, M., & Al-Khatib, O. (2025). Ensemble learning for NIDS based on correlation and embedded feature selection. *Computers*, 14(3), Article 82. <https://doi.org/10.3390/computers14030082>
- [19] Guerra, L., Piras, L., & Giacinto, G. (2025). *GraphIDS: Unifying GNNs and transformers for self-supervised intrusion detection*. arXiv. <https://doi.org/10.48550/arXiv.2509.16625>
- [20] Soltani, N., Nejadshamsi, S., Abou El Houda, Z., Khoury, R., Costa, K. A. P., Falk, T. H., & Avila, A. R. (2026). *Enhancing network intrusion detection systems: A multi-layer ensemble approach to mitigate adversarial attacks*. arXiv. <https://arxiv.org/abs/2603.10413>

- [21] Syrigos, I., Koukoulis, I., & Korakis, T. (2025). *Self-supervised transformer-based contrastive learning for intrusion detection systems*. arXiv. <https://doi.org/10.48550/arXiv.2505.08816>
- [22] Mayhoub, S., Foh, C. H., Mashhadi, M. B., Shojafar, M., & Tafazolli, R. (2026). Talk like a packet: Rethinking network traffic analysis with transformer foundation models. *IEEE Communications Magazine*, 64(5), 36–42. <https://doi.org/10.1109/MCOM.001.2500414>
- [23] Pérez-Jove, R., Munteanu, C. R., Dorado, J., Pazos, A., & Vázquez-Naya, J. (2026). Network traffic foundation models: A systematic review. *Computer Networks*, 276, Article 111998. <https://doi.org/10.1016/j.comnet.2026.111998>
- [24] Zhang, J., Bu, H., Wen, H., Liu, Y., Fei, H., Xi, R., Li, L., Yang, Y., Zhu, H., & Meng, D. (2025). When LLMs meet cybersecurity: A systematic literature review. *Cybersecurity*, 8, Article 55. <https://doi.org/10.1186/s42400-025-00361-w>
- [25] Feng, Y., & Sakurai, K. (2025). *Network intrusion detection: Evolution from conventional approaches to LLM collaboration and emerging risks*. arXiv. <https://arxiv.org/abs/2510.23313>
- [26] Kheddar, H. (2025). Transformers and large language models for efficient intrusion detection systems: A comprehensive survey. *Information Fusion*, 124, Article 103347. <https://doi.org/10.1016/j.inffus.2025.103347>
- [27] Cui, T., Lin, X., Li, S., Chen, M., Yin, Q., Li, Q., & Xu, K. (2025). *TrafficLLM: Enhancing large language models for network traffic analysis with generic traffic representation*. arXiv. <https://arxiv.org/abs/2504.04222>
- [28] Lee, J. Y., & Kim, H. C. (2026). Scale, structure, and stability: When does LLM-based data augmentation improve temporal robustness in web intrusion detection? *Electronics*, 15(7), Article 1344. <https://doi.org/10.3390/electronics15071344>
- [29] Kampourakis, K. E., Kampourakis, V., Chatzoglou, E., Kambourakis, G., & Gritzalis, S. (2026). *Knowledge-to-data: LLM-driven synthesis of structured network traffic for testbed-free IDS evaluation*. arXiv. <https://arxiv.org/abs/2601.05022>
- [30] Otokwala, U. (2024). *Enhancing intrusion detection through data perturbation augmentation strategy* [Research report]. Robert Gordon University.
- [31] Zhu, Y., Kellermann, A., Gupta, A., Li, P., Fang, R., Bindu, R., & Kang, D. (2026). Teams of LLM agents can exploit zero-day vulnerabilities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EACL)*. <https://doi.org/10.18653/v1/2026.eacl-long.2>
- [32] Deng, G., Liu, Y., Li, Y., Yang, R., Xie, X., Zhang, J., Qiu, H., & Zhang, T. (2026). *What makes a good LLM agent for real-world penetration testing?* arXiv. <https://arxiv.org/abs/2602.17622>
- [33] Yang, J., Zhang, R., Shang, C., Li, M., Shang, X., Gao, J., Zhang, Y., & Lu, Y. (2026). Decision boundary-aware generation for long-tailed learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- [34] Sun, Z., Pergola, G., & He, Y. (2026). Boundary-aware LLM augmentation for low-resource event argument extraction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (pp. 4937–4953). <https://doi.org/10.18653/v1/2026.eacl-long.230>
- [35] Deng, J., Shen, Y., Shi, X., & Chai, J. (2026). *DeMix: Debugging training data with mixed data error types by investigating influence vectors*. arXiv. <https://arxiv.org/abs/2606.11616>
- [36] Arik, S. Ö., & Pfister, T. (2021). TabNet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 6679–6687). <https://doi.org/10.1609/aaai.v35i8.16846>
- [37] Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems (Volume 35)* (pp. 507–520). Curran Associates.

Appendix A. Diagnosis interface

The LLM is asked to return a JSON object with three fields. `diagnosis` contains `confusion_patterns`, each with `pattern_type`, `description` and `affected_error_type`; `root_causes`, each with `cause_type`, `description` and a confidence in $[0,1]$; and `suggested_hard_samples`. `perturbation_instructions` contains a list of objects with `target_features` (3 to 5 names), `direction_per_feature`, `magnitude_per_feature` (in units of the feature's standard deviation, clipped to $[0.1,0.5]$), `perturbation_rationale` and `n_variants`. The input is a table of 50 boundary samples rendered as their top-15 SHAP-ranked feature values together with the predicted label, the true label, the confidence and the ensemble disagreement.

The following is an unedited excerpt of the diagnosis returned for UNSW-NB15, round 1, and the instruction that followed from it. It is reproduced verbatim to make the failure in Table 12 auditable.

```
confusion_patterns[0]
```

```
pattern_type      : feature_value_overlap
affected_error_type: both
description       : Many FP and FN samples share similar high values for
                    features like dttl, dwin, swin, sttl, and often
                    similar values for TCP-related features (synack,
                    tcprtt, ackdat). These features, while important, are
                    not sufficient to distinguish benign from attack in
                    boundary cases.
```

```
root_causes[0]
```

```
cause_type : class_overlap_in_feature_space
confidence : 0.9
description: Benign and attack flows share similar distributions for key
                    features such as TCP handshake timings (synack, tcprtt),
                    window sizes (dwin, swin), and TTL values (dttl, sttl).
                    This overlap causes the model to misclassify at the boundary.
```

```
perturbation_instructions[0]
```

```
target_features      : [dttl, dwin, swin, synack, tcprtt]
direction_per_feature: {dttl: decrease, dwin: decrease, swin: decrease,
                        synack: increase, tcprtt: increase}
magnitude_per_feature: {dttl: 0.4, dwin: 0.3, swin: 0.3, synack: 0.2,
                        tcprtt: 0.2}
n_variants           : 8
rationale             : Decreasing TTL and window sizes while increasing TCP
```

handshake timings can push ambiguous samples toward the attack region, clarifying the boundary by simulating stealthier, slower connections typical of attacks.

Appendix B. Reproduction details

Base models. Random Forest: 100 estimators, max_depth 15. LightGBM: 100 estimators, max_depth 10. XGBoost: 100 estimators, max_depth 8. Combination by majority voting. Training uses a 70/30 stratified split with fixed seeds.

DB-HSM. Boundary budget $k = 50$, split evenly between false positives and false negatives; $\alpha = 0.5$ in Equation (1); 3 to 5 target features per instruction; magnitude clipped to $[0.1, 0.5]$ standard deviations; $R = 5$ to 10 variants per sample; maximum rounds $T = 2$; stopping threshold $\theta = 0.0005$. All LLM responses are cached by (round, dataset).

Significance study. Seeds 42, 123, 456, 789 and 1024. Both single-round and multi-round pipelines are run end to end per seed, and the paired delta against the matching baseline is tested with a paired t -test and a one-sided Wilcoxon signed-rank test.