# When Calibrated Detectors Meet New Attacks: Per-Category Reliability of Machine-Learning Intrusion Detection Under Distribution Shift

Khalid Alalawi
College of Computer Science and Engineering (CCSE), Taibah University, Medina 42353, Saudi Arabia

***Abstract*—Machine-learning intrusion detectors are usually reported with accuracy or F1 on a single train and test split, and their confidence scores are often read operationally as probabilities without an explicit calibration check. We test that assumption. We measure the reliability of the predicted probabilities of three classifiers, logistic regression, random forest, and XGBoost, on two benchmarks, NSL-KDD and UNSW-NB15, in two settings: when training and test data share a distribution, and under the shift each benchmark's official split already contains. In distribution, all three detectors are well calibrated, with top-label expected calibration error at most 0.03. Under shift, the outcome depends on its content. On NSL-KDD, whose test set contains attack types absent from training, top-label calibration error rises from at most 0.007 in distribution to between 0.170 and 0.209, and the error is largest on the benign and Remote-to-Local classes, so unfamiliar attacks are labeled normal with high confidence. On UNSW-NB15, whose split keeps the same attack categories, the error stays far smaller, between 0.025 and 0.068, even though a domain classifier detects a clear shift on both benchmarks. Recalibrating on training-distribution data does not reliably transfer to the shifted test set. Supervised target-domain recalibration on a held-out labeled sample from the shifted distribution substantially reduces the calibration error and recovers much of the minority-class recall on NSL-KDD. Most of the aggregate top-label ECE reduction is obtained with about one hundred labeled target samples. This remedy requires labeled observations from the shifted environment and therefore describes reliability after target labels become available, not reliability against attacks that remain entirely unseen. A decision-curve analysis and expected-cost comparison show that an uncalibrated score used at its nominal cost threshold is very costly, and that both recalibration and a target-tuned threshold reduce operating cost, with recalibration additionally improving the multiclass probabilities. A leave-one-attack-out study reproduces the failure and indicates that the attack family for which performance degrades substantially is the one least separable from benign traffic, not the one most distinct from the other attacks.**

***Keywords*—Intrusion detection; probability calibration; distribution shift; expected calibration error; decision curve analysis; class imbalance; network security***

## I. Introduction

Network intrusion detection is now built mostly on supervised machine learning. Detectors trained on flow records reach high accuracy on the public benchmarks, and reported accuracy above ninety-five percent is common [1], [2], [3]. Accuracy is rarely the property that decides whether a detector can be deployed. A detector returns a label and a score, and an analyst has to decide which alerts to act on and in what order. That decision uses the score as a probability. If the score reads 0.9, the operator treats the flow as very likely to be an attack. If it reads 0.5, the operator treats it as uncertain. A probability that does not mean what it says leads to wrong triage, to thresholds set in the wrong place, and to alert volumes that do not reflect real risk.

The property that makes a probability mean what it says is calibration. A detector is calibrated when, among the events it assigns probability 0.8, about eighty percent are attacks. Calibration is separate from accuracy. A model can rank cases well, and so score high on the area under the receiver operating characteristic curve, while its probabilities are systematically too high or too low [4], [5]. Calibration and predictive uncertainty have been studied at length for modern deep networks [4], [6], [7], [8], and calibration is known to degrade when test-time data differ from training data [9]. In intrusion detection, this has begun to receive attention, including work on uncertainty and out-of-distribution behavior [10], calibrated cost-sensitive alerting [11], and concept drift [12], but it is still far less standard than accuracy reporting, and where probabilities are reported, they are seldom checked per class or under shift.

Two facts make this gap matter. First, intrusion detection datasets are heavily imbalanced [13], so aggregate figures are dominated by the common classes and hide the behavior of the rare ones [14], [15]. Second, and more importantly here, deployment is not the setting in which these detectors are usually evaluated. A model in production meets traffic and attacks that were not in its training data, which is a distribution shift. The standard benchmark protocol trains and tests on a single split of one dataset, so it reports the behavior of a detector on data that resemble its training data. Whether the calibration measured that way survives the shift a real deployment faces is the question this study reviews.

We study it directly, training three standard classifiers on two benchmarks and measuring their calibration in two regimes. The first is in-distribution: a held-out split of the training file serves as the test set. The second is shifted: the benchmark's official test file is used instead. Calibration is reported per attack category, not in aggregate alone. Two further questions follow: whether standard recalibration repairs any failure we find, and what that failure costs the decisions an operator makes.

The results are as follows. In distribution, all three detectors are already well calibrated, so the calibration problem is invisible under the usual protocol. Under shift, the outcome

depends on the kind of shift. When the shifted test set contains attack types absent from training, calibration degrades sharply, and the error concentrates on the benign class, so novel attacks are confidently passed as normal. When the shift keeps the same attack categories, calibration degrades only mildly, even though a domain classifier detects a clear shift in both cases. Recalibrating on training-distribution data does not fix the failure. Recalibrating on a held-out labeled sample from the shifted distribution does, and it recovers minority-class recall.

This study makes four contributions. The first is to show that the good calibration of machine-learning intrusion detectors is a property of same-distribution testing and does not carry over to the shift introduced by deployment. The second contrasts two kinds of shift and finds that the kind involving unfamiliar attacks, rather than measured distance between distributions, is associated with failure, and supports this with a controlled leave-one-attack-out study. The third evaluates supervised target-domain recalibration on a held-out labeled sample from the target environment and measures how calibration changes as the labeled sample is reduced. The fourth uses decision-curve analysis and an expected-cost comparison to separate the operating benefit of calibration from that of threshold placement. The remainder reviews related work in Section II, describes data and methods in Section III, reports results in Section IV, discusses them in Section V, and concludes in Section VI.

## II. Related Work

### A. Machine Learning for Network Intrusion Detection

Supervised classification of network flows has been the dominant approach to intrusion detection for two decades, and several surveys map the methods and datasets in use [1], [2], [16]. Recent systematic reviews and a meta-analysis extend this to current machine- and deep-learning methods, datasets, and validation practice [15], [17], [18]. NSL-KDD [19] and UNSW-NB15 [20], [21] are among the most used benchmarks, and CIC-IDS2017 [22] is a common alternative. Ring et al. [13] review the available datasets and their properties. On tabular flow features, tree ensembles lead most comparisons, with random forests [23] and gradient-boosted trees such as XGBoost [24] as the usual strong baselines, and logistic regression a common linear reference, using scikit-learn [25]. Class imbalance recurs across these datasets and is handled with resampling, for example, the synthetic oversampling of Chawla et al. [14], or with cost-sensitive training [26], [27]. This body of work reports detection quality with accuracy, F1, or the area under the receiver operating characteristic curve. The trustworthiness of the predicted probabilities is not usually part of the evaluation, and robustness concerns such as adversarial evasion are treated separately again [28].

### B. Confidence Calibration and Predictive Uncertainty

A predicted probability is calibrated when it matches the observed frequency of the event. Platt scaling fits a logistic function to a classifier's scores [29], and isotonic regression fits a monotone step function without a parametric form [30]. Niculescu-Mizil and Caruana [5] found that maximum-margin methods such as boosted trees and support vector machines push their predicted probabilities away from 0 and 1, giving a characteristic sigmoid distortion, while naive Bayes has the opposite bias and pushes them toward 0 and 1. Guo et al. [4] showed that modern neural networks are badly miscalibrated and that a single temperature on the logits restores calibration cheaply. Minderer et al. [7] revisited this for newer architectures and found that several recent models, in particular non-convolutional ones, can be both accurate and well calibrated, which tempers that trend, and Kull et al. [6] extended parametric calibration to the multiclass case. The quality of a probability is measured with the expected calibration error [31], for which debiased estimators exist [32], and the Brier score [33], [34]. These calibration tools sit within a wider effort to quantify predictive uncertainty in deep networks, which [8] surveys. These tools were developed and are mostly applied to fixed models on same-distribution test data.

### C. Distribution Shift and Out-of-Distribution Behavior

A model deployed in the field meets data drawn from a different distribution than its training data, a situation studied under dataset shift and concept drift [12], [35], [36], [37]. Ovadia et al. [9] measured predictive uncertainty under shift and found that accuracy and calibration both degrade as the shift grows, and that methods calibrated in distribution do not stay calibrated once the data move. Minderer et al. [7] later reported the same decay for recent image classifiers, though some architectures resisted it and in-distribution calibration tended to track out-of-distribution calibration. Detecting that a shift has occurred is itself a task. Rabanser et al. [38] evaluated methods for it, one of which trains a classifier to separate source from target samples, a device grounded in the domain-divergence theory of Ben-David et al. [39]. Hendrycks and Gimpel [40] studied flagging inputs a model should not be confident about. In intrusion detection, the shift is concrete, because new attacks and new traffic appear after a detector is trained. Recent work addresses this directly: Talpini et al. [10] study uncertainty quantification and out-of-distribution detection for ML-based intrusion detection and show conventional networks assign high confidence to unknown-class inputs, and Shyaa et al. [12] survey concept- and feature-drift-aware detection. The guidance of Arp et al. [41] on evaluation pitfalls in security machine learning is still often not followed.

### D. Decision-Analytic Evaluation

Whether a probability is useful depends on the decision it informs and on the relative cost of the two kinds of error. Cost-sensitive learning weights errors by their cost [27]. Decision curve analysis, introduced by Vickers and Elkin [42] and set out further by Vickers et al. [43], evaluates a model by its net benefit across the operating thresholds an operator might use, and needs no external cost figure specified in advance. It is standard in clinical prediction. Closest to our operating-cost analysis, Youssef [11] presents a streaming intrusion detector that maps a change-point posterior through isotonic calibration to a calibrated conditional attack probability and derives its alerting threshold from operator-specified costs and budgets, reporting isotonic-calibration Brier-score reductions across several benchmarks. That work targets a single binary alerting decision on streaming data. We complement it with a per-category multiclass calibration analysis under an official unseen-attack split, and use decision-curve analysis to separate the operating value of calibration from that of threshold placement. Most

directly, Ganesh et al. [44] build a two-stage generative-AI detector for cloud security and evaluate its calibration on NSL-KDD and UNSW-NB15, comparing Platt, isotonic, Beta, and temperature scaling by expected calibration error and reporting per-class calibration curves and decision-curve analysis, the closest prior application of these tools to intrusion detection on these benchmarks.

### E. Positioning

These lines of work are converging but remain partial. Intrusion-detection studies report detection quality and, increasingly, uncertainty, but rarely the per-class reliability of the probabilities under an official unseen-attack split. The most closely related work [44] compares these calibration methods and reports decision-curve analysis on the same benchmarks, but it fits its calibrators on a training-derived validation set and reports test-set calibration as one part of evaluating a proposed detector. It does not contrast in-distribution against shifted calibration, compare recalibration fitted on the source against recalibration fitted on a target sample, or report per-category reliability under the shift. The calibration literature measures reliability on fixed same-distribution data. The shift literature shows calibration is fragile under shift, largely on image and text models, without separating the kinds of shift a security setting produces. This study measures the calibration of intrusion detectors per attack category, in distribution and under the shift the benchmarks encode, distinguishes shift that introduces new attacks from shift that does not, tests supervised target-domain recalibration and measures how its results change as the labeled target sample is reduced, and expresses the result as an operating cost, separating the contribution of calibration from that of threshold adaptation. Post-hoc calibration also rests on assumptions that matter under shift. A calibrator fitted on source-domain validation data depends on the score-label relationship transferring to the target domain, an assumption that can fail when the distribution changes [9]. Target-domain fitting avoids that source-to-target transfer assumption but requires labeled observations from the shifted environment. Class-wise isotonic regression can also have little data from which to estimate a mapping for very rare classes. Threshold tuning can lower binary operating cost without correcting the multiclass probability vector, so calibration and threshold adaptation are evaluated separately here.

## III. Materials and Methods

### A. Datasets and Shift Protocol

We use two public benchmarks. NSL-KDD [19] is the cleaned successor to KDD Cup 1999. Its training file holds 125,973 records and its official test file holds 22,544 records, each with forty-one features. Three of these features, the protocol, service, and flag fields, are categorical. Attack labels are grouped into the four standard categories: Denial-of-Service, Probe, Remote-to-Local, and User-to-Root, which, with normal traffic, gives five classes. UNSW-NB15 [20] provides forty-nine flow and packet features. We use the partitioned release analyzed by Moustafa and Slay [21], which holds 175,341 training and 82,332 test records over forty-two analysis features, three of them categorical (the protocol, service, and state fields), across ten classes, one benign and nine attack families. Both are heavily imbalanced. In NSL-KDD, the rarest class, User-to-Root, has sixty-seven test instances by our count of the official test file. In UNSW-NB15, the rarest, Worms, has forty-four.

The official train and test partitions of these benchmarks are not two samples of one distribution. NSL-KDD places attack types in the test set that do not occur in training, so its split contains a shift by construction, and we refer to this as an unseen-attack shift. UNSW-NB15 keeps the same attack categories on both sides but re-partitions the traffic, a milder shared-taxonomy shift. Moustafa and Slay [21] report that these official partitions share the same distribution, non-linear and non-normal, on a Kolmogorov-Smirnov, skewness, and kurtosis analysis, so the separability our domain classifier finds on UNSW-NB15 reflects feature-space distance within a split its designers describe as matched, and it is against that baseline that the mild calibration change we observe on this benchmark should be read. We use both splits to create two regimes. For the in-distribution regime, we ignore the official test file and split the training file into three stratified parts: sixty percent for training, twenty percent for a calibration set, and twenty percent for an in-distribution test set. For the shifted regime, we train on the sixty-percent portion and test on the official test file. To study supervised target-domain recalibration, we set aside a stratified twenty-five percent slice of the official test file as a labeled calibration sample and evaluate on the remaining seventy-five percent. This is a supervised adaptation protocol. The calibrator has access to labels from the shifted distribution, while the remaining seventy-five percent is kept separate for evaluation.

### B. Classifiers and Preprocessing

We compare three classifiers spanning the families common in intrusion detection: logistic regression as a linear reference, a random forest of one hundred trees with a maximum depth of fifteen, and XGBoost with one hundred trees of maximum depth six and a learning rate of 0.1, using the histogram method. The settings are fixed rather than tuned, so the comparison reflects the effect of shift and recalibration rather than a hyperparameter search. Numeric features are standardized, and the three categorical fields are one-hot encoded, with categories seen fewer than twenty times folded into an infrequent group to keep the feature count bounded. The scaler and encoder are fitted on the training portion alone and applied without refitting to every other split, so no information from the calibration or test data reaches the model. This split-first order is the standard guard against the leakage that Arp et al. [41] identify as a frequent error.

### C. Calibration Methods

We apply three post-hoc calibrators, each fitted on a held-out calibration set and applied to a separate evaluation set. Temperature scaling divides the logits by a single positive scalar chosen to minimize the negative log-likelihood on the calibration set [4], which softens or sharpens every probability without changing the ranking. Platt scaling fits a logistic function to each class score [29], applied here in a one-versus-rest decomposition [30]. Isotonic regression fits a monotone step function to each class score in the same scheme [30]. For the one-versus-rest calibrators, the per-class outputs are renormalized to sum to one. All three are fit on the whole calibration set without an internal cross-validation loop, which

makes the results deterministic given the data and the software versions we report. Temperature scaling operates on logits, which we take as the decision scores for logistic regression, the native margins for XGBoost, and the log of the clipped class probabilities for the random forest, which has no native logits.

### D. Reliability Metrics

We measure calibration with the expected calibration error and the Brier score. The expected calibration error bins predictions and averages the absolute gap between mean confidence and accuracy within each bin [31], using fifteen equal-mass bins. The reliability diagrams are drawn with twelve bins. We report two forms. The top-label form uses the predicted class and its confidence, the usual multiclass definition. The class-wise form averages, over classes, the one-versus-rest error for each class, exposing miscalibration in a class even when that class is rarely the top prediction [6], [32]. The tabulated values use fifteen equal-mass bins, so each bin holds the same number of predictions. We also report the per-category one-versus-rest error. The Brier score we report is the total multiclass score: for each example, the sum over classes of the squared differences between the predicted probabilities and the one-hot label, averaged over examples [33]. Unlike ECE, it does not depend on confidence bins, so it provides a binning-independent check on overall probability quality. Because it reflects both calibration and refinement [34], it is read alongside ECE and macro-F1 rather than as a pure calibration measure.

### E. Shift Severity

To measure how far the shifted test set stands from the training distribution, independently of any detector, we train a domain classifier to separate in-distribution samples from shifted samples and report the area under its receiver operating characteristic curve from three-fold cross-validation. A value near 0.5 means the two are indistinguishable, and a value near one means they are fully separable [38], [39]. This gives a single detector-independent number for the size of each shift. It measures how far apart the two distributions are, not which features differ or whether the conditional label distribution changes, and we treat values across datasets as indicative rather than directly comparable. Table I summarizes the two datasets and the measured severity of each shift. The domain classifier is a random forest with 100 trees and a maximum depth of 12. Its inputs use the training-fitted scaler and one-hot encoder described in Section III-B. For each seed, 5,000 observations are sampled without replacement from each domain to form a balanced 10,000-record dataset. AUROC is averaged over three stratified folds with shuffling, with the same seed used for sampling, the classifier, and cross-validation.

### F. Decision-Curve Analysis and Expected Cost

To express calibration in operational terms, we reduce the problem to attack versus benign and vary the operating threshold. For a threshold t, a flow is flagged when its attack probability is at least t, and the net benefit is TP/N minus (FP/N) times the odds t/(1-t), where N is the number of evaluated records [42], [43]. We plot net benefit against t for the uncalibrated and the recalibrated detector, alongside the reference policy of flagging every flow. We also report an expected cost. Missing an attack is treated as r times as costly as a false alarm. The expected cost is (r times the false negatives plus the false positives) divided by N. To separate the value of calibration from that of threshold placement, we report four operating conditions at each r: the uncalibrated score at a fixed threshold of 0.5, the uncalibrated score at the decision-theoretic threshold 1/(1+r) [27], the uncalibrated score at a threshold tuned to minimize cost on the same labeled target sample used for recalibration, and the target-recalibrated probability at the decision-theoretic threshold. We report r equal to five, ten, and twenty.

### G. Leave-One-Attack-Out Protocol

To test whether unfamiliar attacks are associated with calibration degradation, we build a controlled shift on UNSW-NB15. We remove one attack family from the training data so it is absent at training time and present, as a novel family, in the test set, and we measure the binary attack-versus-benign detector's recall on that family and the mean confidence with which it assigns the family to the benign class, on a held-out evaluation subset, using XGBoost with two hundred trees as the binary detector. We repeat this for several families. To characterize each family, we compute two feature-space separability scores, each the area under a random-forest classifier's curve under three-fold cross-validation with preprocessing fitted inside each fold: how separable the family is from the other attack families, and how separable it is from benign traffic. A low family-versus-benign score means the family resembles normal traffic.

### H. Reproducibility

The core calibration metrics and the per-category results are reported as mean ± standard deviation (SD) over three seeds, 42, 7, and 123, which re-partition the training data and so give genuine seed variability. The target-sample-size sweep draws ten stratified labeled subsamples at each of the three seeds, thirty draws per size, and is reported as the mean with a ninety-five percent confidence interval. The uncalibrated baseline has no calibration subset to resample and is summarized as the mean and standard deviation over the three seeds. The reliability diagram and the per-category and decision-curve figures use seed 42 as an illustrative run. In the leave-one-attack-out study, the withheld and full models are fitted deterministically on the full training data, so the three seeds vary only the evaluation partition, giving the small spread reported. The calibrators are deterministic given the data. Both datasets are public [19], [20]. The experiments were run under scikit-learn 1.8.0 [25] and XGBoost 3.3.0 [24] with fixed data partitions and software versions.

TABLE I. DATASETS AND MEASURED SHIFT SEVERITY

| Dataset | Classes | Train / Test | Shift AUROC |
|---|---|---|---|
| NSL-KDD | 5 | 126.0k / 22.5k | 0.87 |
| UNSW-NB15 | 10 | 175.3k / 82.3k | 0.81 |

## IV. RESULTS

### A. In Distribution, the Detectors are Already Calibrated

When the calibration and test sets are drawn from the training distribution, all three detectors are well calibrated on both benchmarks. On NSL-KDD, the top-label expected calibration error is 0.000 for XGBoost, 0.006 for the random

forest, and 0.007 for logistic regression. On UNSW-NB15, it is 0.011, 0.030, and 0.012 for the same three. These values are small, and they are what a study that trains and tests on one split would report. Under the usual protocol, there is no calibration problem to see.

### B. *New-Attack Shift Substantially Degrades Calibration, While Same-Category Shift Degrades it Far Less*

The shift the official splits contain changes this, and not in the same way for the two benchmarks. Table II reports the shifted regime. On NSL-KDD, the top-label error rises to between 0.170 and 0.209 for every model, about thirty times the in-distribution value for logistic regression and the random forest, and from a value near zero for XGBoost. On UNSW-NB15, it stays far smaller, between 0.025 and 0.068, and the change is model-dependent, decreasing slightly for the random forest while rising modestly for the other two. A domain classifier separates in-distribution from shifted samples with an area under the curve of 0.87 on NSL-KDD and 0.81 on UNSW-NB15. The score is somewhat higher for NSL-KDD, but because the two values arise from different feature spaces, they describe source-target separability rather than giving directly comparable shift magnitudes. The difference in calibration degradation between the two benchmarks is far larger than the difference between these separability scores. What differs is the content of the shift: the NSL-KDD test set contains attack types absent from training, whereas the UNSW-NB15 split keeps the same attack categories.

The failure on NSL-KDD is not spread evenly across the classes. Table III gives the per-category error and recall for XGBoost. The error is largest on the benign class, at 0.203, and on Remote-to-Local, at 0.116, while the other categories stay low. This is the security concern in the result: under a shift that brings unfamiliar attacks, the detector hands many of them to the benign class with high confidence, and its recall on Remote-to-Local is only 0.08. A reliability diagram makes the over-confidence visible. In Fig. 1, the in-distribution points lie on the diagonal, and the shifted uncalibrated points fall well below it, meaning the accuracy in each confidence bin is far lower than the confidence. Fig. 2 places the two benchmarks side by side.

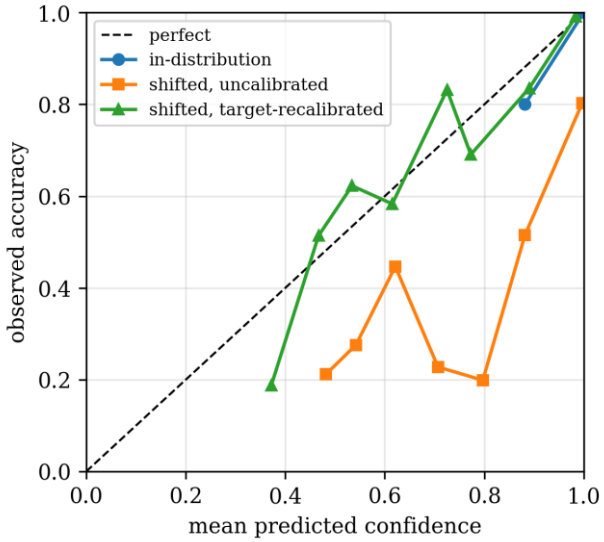


Fig. 1. Reliability of XGBoost on NSL-KDD (illustrative seed-42 result). In-distribution points lie on the diagonal; the shifted, uncalibrated points fall below it, indicating over-confidence; target-domain recalibration returns them toward the diagonal.

TABLE II. RELIABILITY AND DETECTION ON THE SHIFTED (OUT-OF-DISTRIBUTION) TEST SET, MEAN ± SD OVER THREE SEEDS

| Data | Model | ECE in-dist | ECE OOD uncal. | ECE OOD temp (in-dist) | ECE OOD iso (target) | Brier OOD uncal. | Brier OOD iso (target) | F1 OOD uncal. | F1 OOD iso (target) |
|---|---|---|---|---|---|---|---|---|---|
| NSL | LR | 0.007±0.001 | 0.209±0.001 | 0.211±0.000 | 0.058±0.020 | 0.443±0.001 | 0.231±0.007 | 0.557±0.013 | 0.713±0.019 |
| NSL | RF | 0.006±0.000 | 0.170±0.003 | 0.205±0.004 | 0.064±0.008 | 0.386±0.004 | 0.207±0.007 | 0.498±0.003 | 0.719±0.039 |
| NSL | XGB | 0.000±0.000 | 0.206±0.003 | 0.208±0.003 | 0.042±0.014 | 0.419±0.005 | 0.212±0.009 | 0.545±0.018 | 0.681±0.044 |
| UNSW | LR | 0.012±0.002 | 0.068±0.002 | 0.069±0.001 | 0.013±0.002 | 0.337±0.001 | 0.280±0.000 | 0.338±0.003 | 0.375±0.011 |
| UNSW | RF | 0.030±0.002 | 0.025±0.002 | 0.039±0.004 | 0.020±0.002 | 0.279±0.001 | 0.221±0.002 | 0.437±0.006 | 0.469±0.012 |
| UNSW | XGB | 0.011±0.001 | 0.043±0.001 | 0.053±0.002 | 0.009±0.002 | 0.280±0.001 | 0.216±0.001 | 0.497±0.020 | 0.519±0.005 |

LR, logistic regression; RF, random forest; XGB, XGBoost; ECE, top-label expected calibration error; OOD, shifted official-test evaluation; temp, temperature scaling; iso, isotonic regression; F1, macro-averaged F1. Values are mean ± SD over three seeds. An SD shown as 0.000 is smaller than 0.0005 at the displayed precision. Calibrators fit on in-distribution data do not repair the shifted failure; calibrators fit on a labeled target-domain sample do.

TABLE III. PER-CATEGORY ECE AND RECALL FOR XGBOOST ON THE SHIFTED NSL-KDD TEST SET, MEAN ± SD OVER THREE SEEDS

| Category | ECE uncal. | ECE iso (target) | Recall unc./cal. |
|---|---|---|---|
| Normal | 0.203±0.002 | 0.031±0.001 | 0.97±0.00/0.95±0.00 |
| DoS | 0.065±0.003 | 0.020±0.004 | 0.83±0.02/0.86±0.01 |
| Probe | 0.034±0.003 | 0.020±0.003 | 0.64±0.02/0.77±0.02 |
| R2L | 0.116±0.002 | 0.043±0.020 | 0.08±0.02/0.55±0.01 |
| U2R | 0.002±0.000 | 0.001±0.000 | 0.10±0.03/0.18±0.17 |

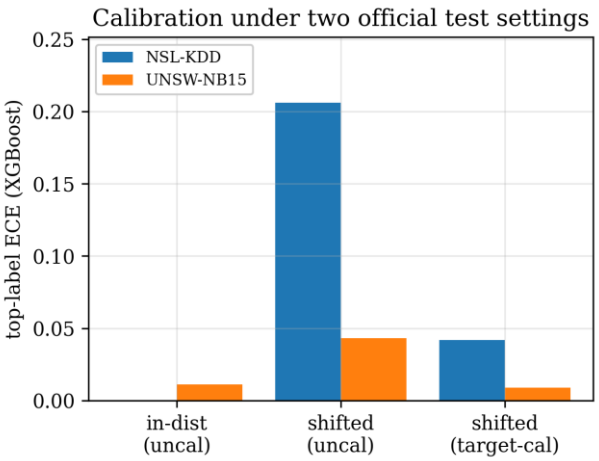


Fig. 2. Top-label ECE for XGBoost in three settings on the two benchmarks (three-seed means). The unseen-attack shift (NSL-KDD) degrades calibration far more than the shared-taxonomy shift (UNSW-NB15), even though a domain classifier finds the source and target readily separable in both.

### C. A Controlled Study Points to Benign-Resembling Families

The contrast between the two benchmarks suggests that greater calibration degradation is associated with unfamiliar attacks rather than with measured source-target separability alone. The leave-one-attack-out study on UNSW-NB15 examines this, and its results are in Table IV. Removing most attack families from training barely changes the detector's ability to flag them: with the family absent from training, recall on Denial-of-Service, Reconnaissance, Backdoor, Shellcode, and Worms stays at or above 0.98, because these families still trip the binary detector. Fuzzers are the exception. With Fuzzers removed from training, its recall falls from 0.90 to 0.40, and the detector assigns Fuzzers to the benign class with a mean confidence of 0.59. The two separability scores help characterize this pattern. Generic feature-distinctness from the other attacks does not explain it: Shellcode (0.99) and Worms (0.99) are more distinct from the other attacks than Fuzzers (0.98), yet do not collapse. What sets Fuzzers apart is that it is the family least separable from benign traffic, with a family-versus-benign area under the curve of 0.94 against 0.997 or above for every other family, and it is the only family the detector confidently labels benign. The pattern is consistent with the NSL-KDD result, where the failing categories, Remote-to-Local and User-to-Root, are those that most resemble normal connections [3]. The margin is modest, and we read the effect as an association between resemblance to benign traffic and confident misclassification rather than a demonstrated mechanism. Ganesh et al. [44] likewise attribute part of the NSL-KDD minority-class difficulty to overlap between these classes and normal traffic. This control is binary and limited to UNSW-NB15, whereas the main NSL-KDD result is multiclass. It supports the observed association but does not establish the mechanism behind the NSL-KDD calibration failure.

TABLE IV. LEAVE-ONE-ATTACK-OUT ON UNSW-NB15 (BINARY ATTACK VS BENIGN), MEAN OVER THREE EVALUATION PARTITIONS

| Withheld family | Rec. in train | Rec. novel | Benign conf. | AUROC vs attacks | AUROC vs benign |
|---|---|---|---|---|---|
| Exploits | 0.99 | 0.98 | 0.07 | 0.93 | 0.998 |
| Fuzzers | 0.90 | 0.40 | 0.59 | 0.98 | 0.942 |
| DoS | 1.00 | 1.00 | 0.02 | 0.89 | 0.999 |
| Reconnaissance | 1.00 | 1.00 | 0.12 | 0.97 | 0.999 |
| Backdoor | 1.00 | 1.00 | 0.01 | 0.84 | 0.999 |
| Shellcode | 0.99 | 0.99 | 0.08 | 0.99 | 0.998 |
| Worms | 1.00 | 1.00 | 0.03 | 0.99 | 0.999 |

Recall on the withheld family when present in training, when novel, and the mean confidence with which the novel family is labeled benign. The last two columns give feature-space separability (leak-free cross-validated AUROC): the novel family versus the other attacks, and versus benign traffic. Only Fuzzers collapse, and it is the least separable from benign traffic rather than the most distinct from other attacks.

### D. Supervised Target-Domain Recalibration Reduces Shifted Calibration Error

A natural response to miscalibration is to fit a calibrator on held-out data and apply it. Table II shows that fitting the calibrator on data from the training distribution does not repair the shifted failure. On NSL-KDD, temperature scaling fitted in distribution leaves the XGBoost error at 0.208 and raises the random-forest error from 0.170 to 0.205, because it sharpens the probabilities in the wrong direction. Supervised target-domain recalibration, fitted on a held-out labeled sample from the shifted distribution, behaves differently. These results describe performance after labeled target observations become available. They do not describe a detector facing new attacks with no labeled target data. Temperature scaling on the target sample reduces the error by roughly one third to one half across the three models, and isotonic regression on the target sample lowers it much further, to 0.042 for XGBoost, 0.064 for the random forest, and 0.058 for logistic regression, while the Brier score falls from about 0.42 to about 0.21. Isotonic regression gives the lowest error in five of the six model-dataset combinations. For the NSL-KDD random forest, Platt scaling on the target sample is slightly lower, at 0.048. Target-fitted calibrators change the area under the curve only marginally or improve it, so ranking is not traded away for calibration. Source-fitted isotonic regression, in contrast, lowers it substantially for the NSL-KDD random forest and XGBoost, further evidence that source-domain calibration does not transfer reliably.

Target-domain isotonic recalibration does more than lower the calibration error. On NSL-KDD, it recovers much of the rare-class detection. For XGBoost, the macro-averaged F1 rises from 0.545 to 0.681, the recall on Remote-to-Local rises from 0.08 to 0.55, and the recall on Probe rises from 0.64 to 0.77, as reported in Table III. The change for User-to-Root is smaller and much less stable, from $0.10 \pm 0.03$ to $0.18 \pm 0.17$ across the three partitions. The full official test file contains only sixty-seven

User-to-Root records, with about fifty in each evaluation slice. The small U2R ECE should be read cautiously at this support. Concretely, the shift makes the benign class overconfident, so many attack flows are given to benign at the argmax. Deflating the benign probability lets the attack classes win, which raises their recall. Because the one-versus-rest isotonic mappings are fitted separately and then renormalized, they can change the relative class probabilities and the argmax prediction. The recall increase reflects this change in the multiclass decision rather than a probability adjustment that preserves class ranking. On UNSW-NB15, where miscalibration was mild, the same recalibration lowers the already-small error further, from 0.043 to 0.009 for XGBoost, with a small gain in F1. Fig. 3 shows the per-category error before and after recalibration on NSL-KDD, concentrated on the classes that were most miscalibrated.

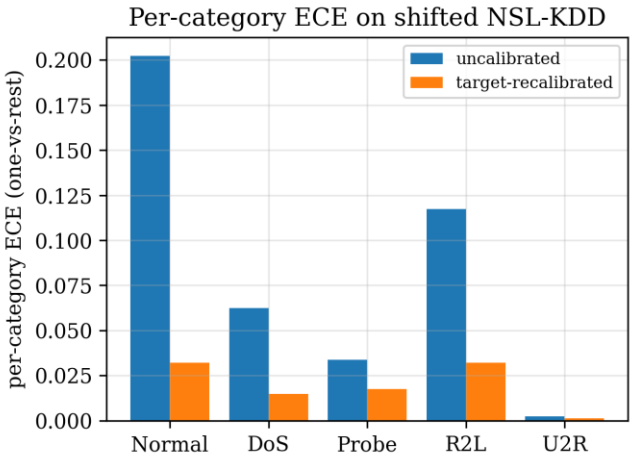


Fig. 3. Per-category ECE on the shifted NSL-KDD test set, before and after target-domain recalibration (illustrative seed-42 result). The reduction is concentrated on the classes that were most miscalibrated.

### E. Effect of Labeled Target Sample Size

Because supervised target-domain recalibration requires labeled data from the shifted distribution, its cost depends on how many such samples are available. Table V reports the top-label error as the labeled sample grows. For each seed, ten stratified subsamples at each size are drawn from the twenty-five percent target-calibration pool, while the remaining seventy-five percent stays fixed for evaluation. Stratification uses the five mapped NSL-KDD classes, not raw attack subtype. At 100 samples, every draw has 43 Normal, 33 DoS, 11 Probe, and 13 R2L records, with no U2R record. Each 100-record draw includes 4 to 8 of the 17 test-only attack types, so none covers every unseen subtype. The thirty resamples and ninety-five percent confidence intervals reflect this composition variation. One hundred labeled samples cut the error from 0.206 to 0.047, a reduction of about three quarters, and lift the macro-averaged F1 from 0.545 to 0.639. Most of the reduction is achieved by one hundred samples. Beyond about five hundred, the error shows no further systematic decrease, varying between 0.032 and 0.040, and the F1 plateaus near 0.68. For aggregate top-label ECE, the 100-sample condition captures most of the observed reduction in this benchmark. It should not be read as reliable per-class calibration at that size because U2R and many unseen subtypes are not represented. The twenty-five percent slice used for the main experiment is not itself small. Only the smaller sample sizes in this sweep are, and we report their values explicitly. For NSL-KDD, this slice contains 5,636 labeled records and represents an optimistic adaptation condition with a large labeled sample rather than a low-label deployment setting.

TABLE V. TARGET-DOMAIN ISOTONIC RECALIBRATION ON NSL-KDD (XGBOOST) VS LABELED TARGET SAMPLE SIZE. NONZERO ROWS: MEAN ± 95% CI OVER THIRTY DRAWS; UNCALIBRATED BASELINE: MEAN ± SD OVER THREE SEEDS

| Labeled target | Top-label ECE | Macro-F1 |
|---|---|---|
| 0 (uncal.) | 0.206 ± 0.003 | 0.545 ± 0.018 |
| 100 | 0.047 ± 0.006 | 0.639 ± 0.007 |
| 250 | 0.037 ± 0.004 | 0.654 ± 0.013 |
| 500 | 0.035 ± 0.003 | 0.669 ± 0.013 |
| 1000 | 0.032 ± 0.003 | 0.668 ± 0.013 |
| 2500 | 0.033 ± 0.004 | 0.681 ± 0.012 |
| 5000 | 0.040 ± 0.004 | 0.677 ± 0.010 |

### F. The Operating Cost, and What Calibration Contributes to it

The final question is what the miscalibration costs an operator, and how much of any benefit comes from calibration rather than from moving the threshold. Table VI reports the expected cost on NSL-KDD for XGBoost and the random forest. Using an uncalibrated score at its nominal decision-theoretic threshold is very costly: for XGBoost at a ratio of ten to one, the expected cost is 1.65, against 0.13 for the target-recalibrated detector at the same threshold, a reduction of 92 percent. That comparison, however, conflates calibration with threshold placement. When the uncalibrated score is instead used with a threshold tuned on the same labeled target sample, its cost falls to 0.13 as well, close to the cost from recalibration. For the random forest, the tuned uncalibrated threshold is lower still. The reading is that most of the operating benefit comes from placing the threshold correctly, which the raw score at its nominal threshold does not, rather than from calibration itself. Calibration's distinct value is twofold: it makes the principled cost-derived threshold usable directly, because the probabilities then mean what they say, and it repairs the multiclass probabilities, recovering the rare-class recall that tuning a single binary threshold does not. The decision curve in Fig. 4 shows the recalibrated detector at or above the uncalibrated one across the evaluated thresholds. On UNSW-NB15, where the shift was mild, all of these differences are small.

TABLE VI. EXPECTED OPERATING COST ON SHIFTED NSL-KDD (ATTACK VS BENIGN), MEAN OVER THREE SEEDS.

| Model | r | uncal. (0.5) | uncal. (pt*) | uncal. (tuned) | cal. (pt*) |
|---|---|---|---|---|---|
| XGB | 5:1 | 0.99 | 0.88 | 0.12 | 0.12 |
| XGB | 10:1 | 1.97 | 1.65 | 0.13 | 0.13 |
| XGB | 20:1 | 3.93 | 3.09 | 0.13 | 0.14 |
| RF | 5:1 | 1.07 | 0.66 | 0.11 | 0.14 |
| RF | 10:1 | 2.12 | 0.82 | 0.11 | 0.18 |
| RF | 20:1 | 4.23 | 0.99 | 0.12 | 0.35 |

Miss costs r times a false alarm. pt* is the decision-theoretic threshold 1/(1+r). The uncalibrated score at pt* is very costly; a threshold tuned on the target sample (uncal. (tuned)) matches or beats target-domain calibration at pt* (cal. (pt*)). Most of the operating benefit is threshold placement, not calibration.

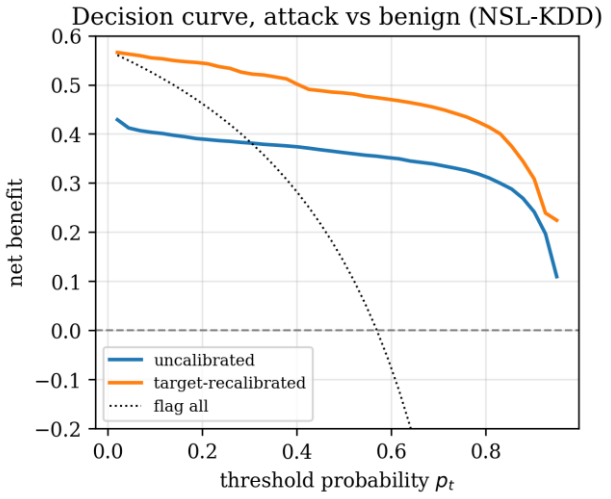


Fig. 4. Decision curve for attack versus benign on NSL-KDD (illustrative seed-42 result). The target-recalibrated detector has net benefit at or above the uncalibrated one across the evaluated operating thresholds.

## V. DISCUSSION

The central finding is that good calibration measured on a same-distribution split did not carry over to the unseen-attack shift represented by NSL-KDD. In distribution, the three detectors were well calibrated, and a study that stopped there would report trustworthy probabilities. Under the shift the benchmarks contain, the probabilities on NSL-KDD were substantially miscalibrated. This is a caution about how intrusion detectors are evaluated. A reliability figure computed on a held-out split of the training distribution can overstate reliability when the evaluation distribution changes, and the evaluation protocol should include unseen attacks.

The second finding refines the first. The domain-classifier score alone does not predict the calibration failure. The observed degradation is more closely associated with the content of the shift. When the shift brought attacks whose behavior was unlike the training attacks, calibration degraded, and the calibration error was concentrated in the benign class. When the shift kept the same attack families, calibration degraded far less, and the change was model-dependent. The leave-one-attack-out study made this concrete: among the withheld families, only Fuzzers collapsed, and it was distinguished not by being the most feature-distinct from other attacks, which it was not, but by being the family least separable from benign traffic. Recalibrating on a labeled target sample did not restore this family: with Fuzzers present in the recalibration sample, its binary recall at the default threshold was 0.34 after recalibration against 0.40 before it, so improving the calibration of the probabilities did not by itself recover detection of the withheld family. Within these benchmark experiments, the observed harm takes a specific form. New attack traffic that is less separable from normal traffic receives confident benign labels. We state this as an association rather than a demonstrated mechanism, since the separability margin is modest and the analysis does not control for every confound.

The results also show what target-domain recalibration can and cannot improve. Recalibrating on training-distribution data did not reliably help and sometimes hurt, so it is not a safe default. Recalibrating on a held-out labeled sample from the target environment did help, with isotonic regression most effective in most cases, and in the severe case it recovered much of the rare-class detection as well as reliability, with about one hundred labeled samples enough for most of the gain. The decision-analytic view adds a caution of its own: most of the reduction in expected operating cost comes from placing the decision threshold correctly, and an uncalibrated score with a threshold tuned on the same target sample matches recalibration on the binary cost. This is consistent with Minderer et al. [7], who report that for practical rejection rates a model's accuracy advantage can outweigh its calibration advantage in a selective-prediction cost analysis. What calibration adds beyond threshold tuning is that the principled cost-derived threshold becomes usable without a search, and that the multiclass probabilities are repaired, recovering rare-class recall. Within the studied benchmark shifts, this supports recalibrating and setting the operating threshold on labeled target data rather than relying on source-domain calibration.

Several limits bound these conclusions. Both benchmarks are built from simulated traffic in a single testbed, so the exact numbers may not carry over to live networks with other traffic mixes and attack tools. The rarest classes are small enough that their per-class figures are noisy, and the calibration of a class with sixty-seven instances in the official test file, such as User-to-Root, cannot be estimated with confidence, a limit we state rather than resolve. The remedy assumes a labeled sample from the target environment can be obtained, which has an operational cost. The hyperparameters were fixed rather than tuned, so the numbers describe untuned defaults. The shift-severity score measures distributional distance, not the direction of the shift, and the family-level separability analysis is descriptive. Finally, we studied two datasets and three model families, and while the pattern was consistent across them, wider replication would strengthen it.

## VI. Conclusion

We measured the reliability of the probabilities produced by three machine-learning intrusion detectors, in distribution and under the shift two standard benchmarks contain, per attack category. In distribution, the detectors were well calibrated. Under a shift that introduced unfamiliar attacks, calibration degraded by a large factor, and the calibration error was concentrated in the benign class, so novel attacks were labeled normal with confidence, while a shift that kept the same attack families left calibration much less affected, even though a domain classifier detects the shift in both cases. Recalibrating on training-distribution data did not repair the failure. Once labeled target observations were available, supervised target-domain recalibration reduced the error, with isotonic regression giving the lowest error in most model-dataset combinations, and it recovered much of the rare-class detection. About one hundred labeled samples captured most of the aggregate top-label ECE gain. A decision-analytic analysis showed that most of the reduction in operating cost comes from correct threshold placement, with calibration contributing a usable principled threshold and repaired multiclass probabilities. The benchmark results support measuring calibration under protocols that include novel attacks, not only on a same-distribution split. Once labeled target observations are available, supervised target-domain recalibration can substantially reduce calibration error and provide data for threshold setting. Whether the same effect sizes hold in live networks remains an open question. A multiclass leave-one-attack-out study on NSL-KDD would test whether the benign-resemblance association persists in the setting where calibration degradation is largest. Future work should also test the protocol on live network traffic and additional IDS benchmarks, compare further model families, and evaluate online recalibration as labeled target observations arrive. The evaluation should account for label delay and annotation cost and test whether selective labeling can reduce the amount of target data needed for recalibration and threshold setting.

## Funding and Conflicts of Interest

This work received no specific grant from any funding agency. The author declares no conflict of interest.

## References

[1] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2016.

[2] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: a systematic study of machine learning and deep learning approaches,” Trans. Emerging Telecommun. Technol., vol. 32, no. 1, e4150, 2021.

[3] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 686–728, 2019.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. 34th Int. Conf. Machine Learning (ICML), 2017, pp. 1321–1330.

[5] A. Niculescu-Mizil and R. Caruana, “Predicting good probabilities with supervised learning,” in Proc. 22nd Int. Conf. Machine Learning (ICML), 2005, pp. 625–632.

[6] M. Kull, M. Perello-Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, “Beyond temperature scaling: obtaining well-calibrated multi-class probabilities with Dirichlet calibration,” in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 12295–12305.

[7] M. Minderer et al., “Revisiting the calibration of modern neural networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 15682–15694.

[8] J. Gawlikowski et al., “A survey of uncertainty in deep neural networks,” Artif. Intell. Rev., vol. 56, suppl. 1, pp. 1513–1589, 2023.

[9] Y. Ovadia et al., “Can you trust your model’s uncertainty? Evaluating predictive uncertainty under dataset shift,” in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 13969–13980.

[10] J. Talpini, F. Sartori, and M. Savi, “Enhancing trustworthiness in ML-based network intrusion detection with uncertainty quantification,” J. Reliable Intelligent Environments, vol. 10, no. 4, pp. 501–520, 2024, doi: 10.1007/s40860-024-00238-8.

[11] M. A. Youssef, “CALIBURN: Operationally Calibrated Streaming Intrusion Detection with Regime-Dependent Conformal Risk Control,” arXiv:2605.24696v2, 2026.

[12] M. A. Shyaa, N. F. Ibrahim, Z. Zainol, R. Abdullah, M. Anbar, and L. Alzubaidi, “Evolving cybersecurity frontiers: a comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems,” Eng. Appl. Artif. Intell., vol. 137, art. 109143, 2024, doi: 10.1016/j.engappai.2024.109143.

[13] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” Comput. Secur., vol. 86, pp. 147–167, 2019.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[15] Y. Zhang, R. C. Muniyandi, and F. Qamar, “A review of deep learning applications in intrusion detection systems: overcoming challenges in spatiotemporal feature extraction and data imbalance,” Appl. Sci., vol. 15, no. 3, art. 1552, 2025, doi: 10.3390/app15031552.

[16] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” Cybersecurity, vol. 2, art. 20, 2019.

[17] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, “A systematic literature review for network intrusion detection system (IDS),” Int. J. Inf. Secur., vol. 22, no. 5, pp. 1125–1162, 2023.

[18] Z. K. Maseer, Q. K. Kadhim, B. Al-Bander, R. Yusof, and A. Saif, “Meta-analysis and systematic review for anomaly network intrusion detection systems: detection methods, dataset, validation methodology, and challenges,” IET Networks, vol. 13, no. 5–6, pp. 339–376, 2024, doi: 10.1049/ntw2.12128.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1–6.

[20] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. Military Communications and Information Systems Conf. (MilCIS), 2015, pp. 1–6.

[21] N. Moustafa and J. Slay, “The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,” Inf. Secur. J., A Global Perspective, vol. 25, no. 1–3, pp. 18–31, 2016.

[22] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116.

[23] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[24] T. Chen and C. Guestrin, “XGBoost: a scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[25] F. Pedregosa et al., “Scikit-learn: machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[26] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009.

[27] C. Elkan, "The foundations of cost-sensitive learning," in Proc. 17th Int. Joint Conf. Artificial Intelligence (IJCAI), 2001, pp. 973–978.

[28] H. Mohammadian, A. A. Ghorbani, and A. H. Lashkari, "A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems," Appl. Soft Comput., vol. 137, art. 110173, 2023.

[29] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers, A. J. Smola et al., Eds. Cambridge, MA: MIT Press, 1999, pp. 61–74.

[30] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2002, pp. 694–699.

[31] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in Proc. 29th AAAI Conf. Artificial Intelligence, 2015, pp. 2901–2907.

[32] A. Kumar, P. Liang, and T. Ma, "Verified uncertainty calibration," in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 3792–3803.

[33] G. W. Brier, "Verification of forecasts expressed in terms of probability," Monthly Weather Rev., vol. 78, no. 1, pp. 1–3, 1950.

[34] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," J. Roy. Statist. Soc. Ser. D, vol. 32, no. 1, pp. 12–22, 1983.

[35] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Eds., Dataset Shift in Machine Learning. Cambridge, MA: MIT Press, 2009.

[36] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," Pattern Recognit., vol. 45, no. 1, pp. 521–530, 2012.

[37] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Comput. Surv., vol. 46, no. 4, art. 44, 2014.

[38] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: an empirical study of methods for detecting dataset shift," in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 1394–1406.

[39] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," Mach. Learn., vol. 79, no. 1–2, pp. 151–175, 2010.

[40] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in Proc. Int. Conf. Learning Representations (ICLR), 2017.

[41] D. Arp et al., "Dos and don'ts of machine learning in computer security," in Proc. 31st USENIX Security Symp., 2022, pp. 3971–3988.

[42] A. J. Vickers and E. B. Elkin, "Decision curve analysis: a novel method for evaluating prediction models," Med. Decis. Making, vol. 26, no. 6, pp. 565–574, 2006.

[43] A. J. Vickers, B. van Calster, and E. W. Steyerberg, "Net benefit approaches to the evaluation of prediction models, molecular markers, and diagnostic tests," BMJ, vol. 352, i6, 2016.

[44] V. Ganesh, N. Khade, H. Taiwade, and P. V. Deshmukh, "A two-stage generative-AI fusion intrusion detection system for calibrated and reliable cloud security," Sci. Rep., vol. 16, art. 20680, 2026, doi: 10.1038/s41598-026-51527-6.