Metrics for Deploying Human-in-the-Loop Classification Models

Pavlo Burda[1][0000−0003−4554−5410], Petros Ioannis Stavroulakis[2],[3][0000−0003−3769−1020], and Sieuwert van Otterloo[1],[4][0009−0004−0249−2122]

[1] ICTI, NL, ictinstitute.nl
pavlo@ictinsitute.nl
[2] Code 4 Thought, GR, code4thought.eu
petros.sta@code4thought.eu
[3] Hellenic Military Academy, Math. & Eng. Science Sector, GR, sse.gr
[4] Hogeschool Utrecht, NL, hu.nl
sieuwert.vanotterloo@hu.nl

Abstract. Due to new legislation and more awareness of AI-related risks, many AI decision models are deployed with human supervision in place where uncertain predictions are reviewed by human experts. However such models are often selected for deployment based on standard metrics like accuracy and precision that implicitly assume every decision is final and automated. When a human reviewer is part of the deployed pipeline, the standard metrics describe a system that is never deployed as such and can rank candidate models differently than a deferral-aware evaluation does, and it might be difficult to discuss the effect and cost of human intervention with diverse stakeholders. We define two performance metrics that are designed to accurately rank model performance by taking this human supervision into account. Based on the selective-classification literature, we propose: a Deferral-Aware-F1 (DAF1), the F1 score computed only over the cases a classifier accepts at a given coverage; and the Cost-Aware-Score (CAS), which weights DAF1 by the relative cost of a human review. Both use a model confidence scores to accept or defer decisions to the human. We use the UNSW-NB15 dataset and the SOC alert-triage scenario to illustrate where human supervision would be required by the AI Act, as it involves potential fraud and blockage of a service. We show how our metrics vs. traditional metrics can change model rankings and error rates and how they can support trade-off discussions across engineering, operations and business stakeholders.

Keywords: Human-in-the-loop · Metrics · Selective classification

1 Introduction

Deciding whether a machine learning (ML) model is good enough to deploy in a real-world application, requires the inclusion of cost and system-level considerations. ML models rarely get deployed in isolation in the real world and usually

come as a part of a system, and are always bound by the hidden real world costs of such applications. Beyond a model's nominal performance metrics, deployed systems are almost never fully autonomous: all or part of their predictions are reviewed by human experts, especially in sensitive domains and on uncertain cases. One such cost is that all or part of their predictions are reviewed by human experts, especially in sensitive domains and on uncertain cases.

In insurance claims and mortgage lending, such reviews are often mandated by regulators [5,4]. More generally, article 22 of the GDPR forbids fully automated decision-making for highly consequential decisions using subject's personal data, therefore warranting a human review [23,6]. The AI Act mandates the human-in-the-loop approach to high-risk AI system. For instance, article 14 of the AI Act requires high-risk AI systems the explainable and allow a human to review a decision [24,19]. Even in domains without such explicit oversight, like network intrusion detection or content moderation, human reviews are still there in practice to avoid costly mistakes and improve the models themselves [7,25].

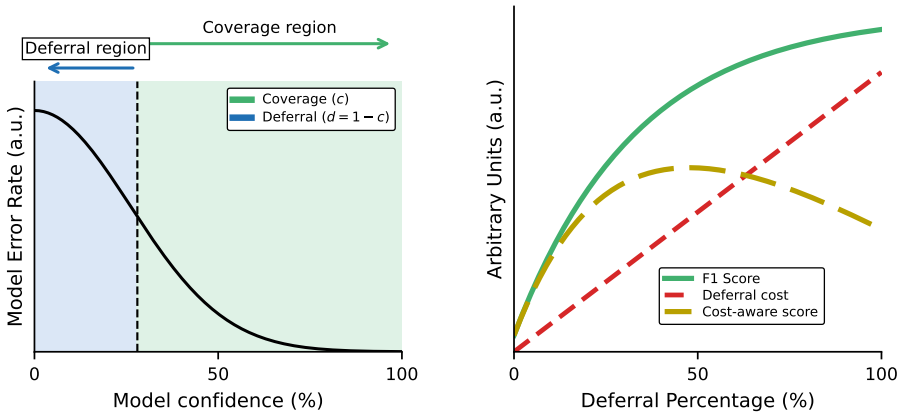
The choice of what to deploy therefore involves several trade-offs at once [12]: should a company deploy an accurate but expensive model to minimize the human interventions? should an explainable, but less performant model be used instead? how should we balance the trade-offs between the deferring decision to human reviewers, model complexity and deployment costs?

Stakeholders with different backgrounds (engineers, product managers, business developers, directors) need to navigate these trade-offs together, often on the basis of model-based performance metrics – like accuracy or precision – that do not take into account the system or cost restrictions. Such metrics are useful for AI engineers but are not always relevant in the boardroom, as they often approximate the real picture and implicitly assume that the model acts completely autonomously. When a human is in fact part of the deployed pipeline, these headline metrics , at best, can hide the model that makes the best use of a given human review budget and, at worst, can rank the wrong model as the optimal one.This raises the central question of this paper: *how can we express classifier performances in a way that reflects the reality of a human-in-the-loop deployment?*

Drawing on the selective classification literature [16], we propose to use perfomance metrics that include the human-in-the-loop calculation:

- *Deferral-Aware-F1* (DAF1): the F1 score computed on the subset of cases a classifier predicts confidently; the remaining, lower-confidence fraction is deferred to a human reviewer;
- *Cost-Aware-Score* (CAS): weights Deferral-Aware-F1 by an explicit human review cost vs. cost of a misclassification at different deferral rates.

Both metrics are illustrated in Figure 1. Figure 1a depicts the deferral region, the coverage region and the confidence of the model vs the models error rate curve. The F1 score in Deferral-Aware-F1 is computed in the coverage region. The DAF1 can be reported as a single headline value at a chosen operating deferral rate, or as a curve that shows how the model's F1 changes as the automated share of decisions moves (depicted in arbitrary units in Figure 1b, continuous



(a) Error rate by a model prediction's confidence: the 'coverage region' is where we want the model to decide, the 'deferral region' is the ideal set of decisions to defer to the human (i.e., 'low confidence').

(b) DAF1 and CAS illustrated: F1 score ideally increases from a baseline value (binary F1) as the deferral rate increase; following an increasing deferral cost, the F1 Cost-Aware Score is penalized.

Fig. 1: DAF1 and CAS illustrated.

green line). The CAS plots a curve that allows to define an optimum operating point at which the classification pipeline should be deployed at a given cost setting estimated in relation to the relative cost of a miscalssification (depicted in arbitrary units in Figure 1b, dashed olive and red lines).

We illustrate both metrics in a Security Operations Center (SOC) scenario where analysts would review uncertain cyber security alerts. To do so, we evaluate four common classifiers – logistic regression, random forest, XGBoost and a simple multi-layer perceptron – on the public UNSW-NB15 intrusion-detection dataset [21]. We show that, compared with nominal binary metrics, the deferral-aware metrics can condition which model a SOC manager should choose. For instance, among four classifiers, a model ranked last under binary F1 ties for the best (logistic regression vs. random forest). We discuss how using the explicit cost assumption and deferral rate can provide a common vocabulary for trade-off discussion across engineering, operations, and business stakeholders.

The rest of the paper is organized as follows. Section 2 introduces DAF1 and CAS. Section 3 describes the UNSW-NB15 dataset and the SOC alert-triage scenario. Section 4 illustrates the use of metrics on classic tabular classifiers. Section 5 discusses use of metrics in practice, limitations and open questions.

2 Deferral-aware metrics

In many real-world settings, machine-learning models are not deployed to act autonomously: their predictions are reviewed by a human expert who has the

final say on uncertain cases. Fraud detection, insurance-claim triage, medical screening, and intrusion detection are all examples where a binary classifier benefits from a third option, "I don't know", alongside "yes" and "no", to allow a human review of uncertain cases.

However models are typically evaluated and ranked using standard binary metrics (Accuracy, Precision, Recall, F1, ROC-AUC, PR-AUC), all of which assume the model decision is final for every case. Whereas developers and data scientists understand the nuances behind these numbers, the non-technical stakeholders responsible for deployment decisions often rely on a single headline metric (Accuracy, Precision or F1 score) that does not account for such nuances and consequently for the deployment reality.

We draw from the selective classification field and propose the use of a Deferral-Aware-F1 (DAF1) to summarize the trade-off of deferral to the human reviewer in a single figure (as a single-headline value or as a curve), and a Cost-Aware-Score (CAS) for the F1 to include the relative costs of human effort in the Deferral-Aware-F1. Both are intended to help evaluate the classification models in real deployment decisions with diverse stakeholders and decision-makers. The DAF1 and CAS reward a model for "correctly abstaining" on borderline cases and summarize the trade-off of the associated human review costs.

The mechanism illustrated in this paper requires that the classifier produce a confidence score for each prediction (a likelihood or a posterior probability). This applies directly to a logistic regression, random forests, multi-layer perceptrons, and any other probabilistic machine learning algorithm. Predictions whose confidence scores below a chosen threshold(s) are rejected by the model and deferred to a human reviewer while the rest are automatically accepted [5]. To simplify the illustrating scenario, we assume an ideal reviewer who classifies every deferred case correctly.

The SOC example. We illustrate both metrics on a network intrusion-detection dataset introduced in Section 3. The example scenario is a Security Operations Center (SOC): a facility that collects and analyzes network traffic from client organizations with the goal of detecting and stopping cyber attacks. Given the high volume of traffic and data, automatic classifiers detect potentially dangerous events and might automatically block connections or user accounts. The typical task of the SOC analyst is to analyze and review such decisions to confirm or deny an event being a real attack. The human-on-the-loop intervention is fundamental because a missed cyber attack or a wrongly blocked connection or user account can have sever consequences on the client organization and their users or customers. For instance, in the effort to block pirated content, the Italian privacy authority blacklisted a domain name responsible for Google Drive services in Italy. This led to a multi-day denial of service across various economical sectors, including some public institutions [3].

In our scenario, the automated detector dispatches high-confidence verdicts automatically, and escalates low-confidence events to a SOC analyst for manual

[5] It is also possible to defer based on anomalies (e.g., PyOD: `https://pyod.dev`).

triage. Analyst time is the scarce resource, and alert fatigue from false positives is the principal operational problem in production SOCs [2], therefore the share of events the model is allowed to escalate to the analyst has a direct operational impact. It is sensible to consider a 10–30% (low-confidence) alerts to be deferred for triage to the analyst. Indeed, this can be the analyst capacity a SOC is reasonably willing to spend [18].

2.1 Deferral-Aware-F1 (DAF1)

The trade-off between predictive performance and rejection rate has been studied in previous work on the optimum error–reject trade-off curve [8]. This subfield is often called *classification with reject option*, *selective classification*, or *learning with rejection* [17,16]. The most common definition and visualization within selective classification are the risk–coverage curve (El-Yaniv and Wiener [10]) and Area Under the Risk–Coverage curve (AURC) (Geifman and El-Yaniv [13]), where the "selection" exchanges automatic classification coverage for an increase in the quality of the automated decisions (improving error rates or accuracy [22]).

Models with rejection mechanisms often estimate an optimal confidence measurement or rejection function based on examples and/or probabilities [16], with an explicit rejection cost [17], a target risk or coverage guarantee [13], or even a learned model of the human expert being deferred to [20,22]. In this paper, we use a simple design where a confidence-based rule decides cases to reject on a trained classifier. Since our object of study is the evaluation metric, we do not design or evaluate a rejection mechanism.

Error rate or accuracy, however, can be misleading in certain situations, like with asymmetric misclassification costs or imbalanced classes. We adopt the F1 score as a 'safe' metric that accounts for a balanced precision and recall[6] in settings where both FPs and FNs should be accounted for, like a SOC. We define selective-F1 by scoring the accepted subset with F1 in place of the error rate in the selective-risk and risk-coverage construction [13,10], and call our headline metric *Deferral-Aware-F1* (DAF1). Similar efforts include precision- and recall-reject curves (Fischer and Wollstadt [11]) which extend the concept of accuracy-rejection curve – used here – to account for imbalanced data cases. DAF1 simply represents the prevision- and recall-reject curves into the single headline number that non-technical stakeholders might benefit from, at the price of less detail.

Setup. Let $\hat{p} \in [0,1]$ be the predicted probability of the classifier that a case is positive, and let $t \in (0,1)$ be a decision threshold[7]. A *selective classifier* extends the binary decision rule ($\hat{p} \geq t$) with the option to abstain: the model only accepts cases where it is confident and *defers* the rest to a human reviewer (the SOC analyst in our example). We use the distance from the threshold, $|\hat{p} - t|$, as the confidence measure: cases far from t are cases where the model is most certain about, while cases near t are uncertain. Given the N test cases sorted by this confidence, the model automates the most-confident $\lfloor cN \rfloor$ of them:

[6] F_1 is $F_{\beta=1}$ by equal weight for recall and precision: $F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$.

[7] In our experiments t is tuned on a calibration split; see Section 4.

- predicting positive class ('attack') if $\hat{p} \geq t$;
- predicting negative class ('not attack') if $\hat{p} < t$;

The remaining $N - \lfloor cN \rfloor$ least-confident cases are *deferred* to the human reviewer (the analyst). The automated fraction $c \in [0, 1]$ is the *coverage*, and $1 - c$ is the share of decisions escalated to the human (deferral rate).

Deferral-Aware-F1. Let $\mathcal{A}(c) \subseteq \{1, \ldots, N\}$ be the set of accepted (automated) cases at coverage c under the rule above. The Deferral-Aware-F1 is the ordinary F1 score restricted to $\mathcal{A}(c)$:

$$F_1(c) \;=\; \frac{2 \cdot \mathrm{TP}_{\mathcal{A}(c)}}{2 \cdot \mathrm{TP}_{\mathcal{A}(c)} + \mathrm{FP}_{\mathcal{A}(c)} + \mathrm{FN}_{\mathcal{A}(c)}},$$

where TP, FP, FN are counted only over the accepted subset at threshold t. Computing the F1 with c from 1 (no deferral, ordinary binary F1) to a given floor k constitutes the Deferral-Aware-F1 curve $F_1(c)$. In selective classification literature, we would call it selective F1 on par with selective-risk [13].

This curve is our metric to evaluate a model accounting for the human review capacity. In our SOC example, each point on the Deferral-Aware-F1 curve, e.g., $F_1(c = 0.95)$, is the F1 the SOC would see on the automated decisions at that coverage (the algorithm deciding on 95% of cases, 5% decided by analysts)[8].

With DAF1, stakeholders can be informed of the actual performance of the model by accounting for the manual reviews. In our example, the shape of the DAF1 curve would reveal how quickly performance on the automated subset improves as borderline cases are escalated to an analyst (see Section 4).

2.2 Cost-Aware-Score (CAS)

The DAF1 curve summarizes the trade-off of deferral to a reviewer but does not consider the relative cost of deferring. A SOC manager cannot decide on given coverage and deferral rate based only on DAF1 but must also consider the operational costs given a coverage. The *Cost-Aware-Score* (CAS) makes this consideration explicit by weighting in the analyst review effort. At a given operating point, c, with Deferral-Aware-F1 $F_1(c)$ and deferral rate $r_{\text{defer}} = 1 - c$,

$$\mathrm{CAS}(\alpha, c) \;=\; F_1(c) \cdot \big(1 - \alpha \cdot r_{\text{defer}}\big),$$

with $\alpha \in [0, 1]$ the deferral penalty rate. We use it as a proxy for the cost of an analyst relative to the cost of a missclassification. The two extremes of α can be interpreted as follows:

- $\alpha = 0$: human review is free. The metric completely ignores the deferral costs with the optimum at lowest coverage, an highest referral rate.

[8] From this definition we can also compute the average F1 across the curve by computing the F1-coverage Area Under Curve (FC-AUC) which is discussed in Appendix B.

– $\alpha = 1$: the score is scaled by the coverage itself, CAS $= F_1(c) \cdot c$, so a fully deferred workload scores zero. The metric penalizes deferrals at their max, and the optimum approaches the binary F1 at the highest possible coverage.

In between, α plots an interpolation with the optimum defined by $c^\star(\alpha) = \arg\max_{c \in [k,1]} \mathrm{CAS}(\alpha, c)$ with k the bottom coverage floor. In the examples of next sections, we illustrate CAS at different levels such as $\alpha \in \{0.1, 0.2, 0.3\}$, corresponding to "review is cheap" (about a tenth of a misclassification), "review is moderate" (a fifth), and "review is expensive" (a third).

Linking cost functions to performance metrics of classifiers with rejection [9], as well as efforts to visualize such trade-offs [14,15] have been addressed in previous work, . These evaluate the classifier together with its rejector and are meant for whoever designs the rejection mechanism. We instead evaluate an already-trained classifier under a fixed, confidence-based deferral rule, for an audience that has to agree on a review budget. Learning-to-defer methods [20,22,26] additionally model the human reviewer; we assume an ideal reviewer (Section 5).

3 Dataset

We use the UNSW-NB15 dataset [21] which is a network intrusion-detection benchmark generated at the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). In the cyber security field, this is a popular benchmark for intrusion detection/prevention systems (IDS/IPS) evaluation. IDS are the main tool used at SOCs to automatically detect and decide if blocking a given connection or activity. The UNSW-NB15 dataset contains synthetic background network traffic mixed with attacks from nine different families, such as shellcode or exploits. It has 42 connection-level features and several derived statistics like byte and packet counts per direction, among others. We use the prepared `UNSW_NB15_training-set.csv` and `UNSW_NB15_testing-set.csv` dataset partitions published with the dataset on Kaggle [9] and combine them into a single $N = 257{,}673$ data frame (175,341 + 82,332 rows), and we cut our own training and testing splits. Table 6 in Appendix A shows the main features of the dataset.

3.1 Prediction task.

Each row is a single network connection. Given the connection-level features, we predict the binary label `label` (1 = 'attack' - block connection, 0 = 'benign' - do nothing). The positive-class prevalence in the Kaggle dataset is 63.91%. The Kaggle version is kept rebalanced upward from the original UNSW-NB15 rate (12% positive-class rate) for simplicity and to support supervised learning [10]. The detector classifies each connection in real time and high-confidence decisions would be dispatched autonomously (e.g., blocking an IP), while low-confidence

[9] Kaggle - `https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15/data`

[10] Appendix C describes the original raw UNSW-NB15 dataset scenario whereby the distribution of model posterior probabilities conditions the thresholding choices.

decisions would are escalated to a SOC analyst for triage. With this typical scenario, we deem the DAF1 coverage and CAS to be valuable metrics to resolve the trade offs of which classification models to select, the costs of analysts' reviews and what performances to expect during the life-cycle of a SOC.

3.2 Dataset preprocessing

We drop the identifier column and we will be using only `label` as the binary target. In the Kaggle distribution there are zero missing values, and NaNs are replaced with 0. Three categorical columns like `proto` required one-hot encoding, reaching 196 total features. Numerical variables were standardized with z-score scaler. The data is split and stratified into 70/15/15 train/calibration/test partitions on the target variable `label` (attack/no attack): 180,371 training, 38,651 calibration, and 38,651 test connections. The 15% calibration split is necessary because the random forest, XGBoost and MLP models we use in Section 4 produce confidence scores that do not represent the real probabilities (like Logistic Regression), and therefore we cannot set our (same) deferral threshold across the model scores. Table 5 in Appendix A summarizes the dataset characteristics after preprocessing.

3.3 A simple example

To explain the definitions of Section 2 we apply them on the UNSW-NB15 dataset with the help of a small example of 10 predictions from a logistic regression on this data. We fix a decision threshold $t = 0.56$ and defer to review the least confident cases first.

Table 1: LR predictions with $|\hat{p} - t|$ confidence and predicted outcomes.

#	$\hat{p}$	$\lvert\hat{p} - t\rvert$	$\hat{y}$	y	**outcome**
1	0.00	0.56	0	0	TN
2	0.00	0.56	0	0	TN
3	0.00	0.56	0	0	TN
4	0.56	0.00	0	0	TN
5	0.56	0.00	0	0	TN
6	0.56	0.00	1	0	FP
7	0.56	0.00	1	0	FP
8	0.56	0.00	1	0	FP
9	1.00	0.44	1	1	TP
10	1.00	0.44	1	1	TP

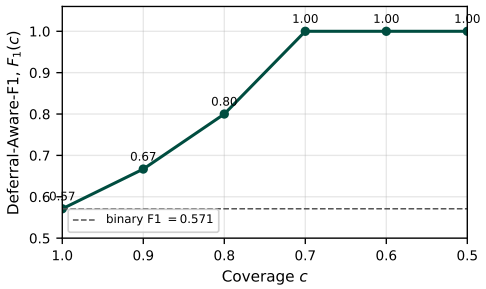


Fig. 2: Deferral-Aware-F1 curve on the 10-case worked example. The dashed line is the binary F1 (0.571).

Table 1 shows the ten predictions with the predicted attack probability $\hat{p}$, the confidence $|\hat{p} - t|$ at $t = 0.56$, and if the prediction ($\hat{y}$) is correct (y). There are

5 TNs, 3 low-confidence FPs (predicted attack but actually benign), and 2 TPs. The standard binary metrics on these ten cases are Accuracy = 0.70, Precision = 0.40, Recall 1.00 and F1 0.57.

Applying the deferral rule to these ten cases at decreasing coverages the Deferral-Aware-F1 curve in Figure 2 takes shape. At full coverage $c = 1.0$ the Deferral-Aware-F1 corresponds to the binary baseline F1 0.57. Already at at $c = 0.9$ (one case deferred, 10%) it goes to 0.67; and at $c = 0.8$ (20%) it gets to 0.80. Finally, at $c = 0.7$ (30% deferred) it reaches the max 1.00.

Assuming that wrong model predictions concentrate in the low-confidence region around the decision threshold, like in Figure 1a, DAF1 grows monotonically because the low-confidence cases are deferred to the human reviewer, who we assume are always correct.

4 Deferral-Aware-F1 and Cost-Aware Score Illustrated

4.1 Illustrating Deferral-Aware-F1

To illustrate the metric on the SOC scenario, we train the four classifiers on the dataset of Section 3: a Logistic Regression (LR), a Random Forest (RF, 200 trees with class-balanced weights), an XGBoost classifier (XGB, gradient-boosted trees at default depth) and a simple Multi-Layer Perceptron (MLP, two hidden layers of 64 and 32 units with early stopping). The models are left at default parameters provided by the `scikit` Python package. In this subsection we mainly refer to the LR model as the motivating example while comparing all the for models.

Decision threshold. Knowing that the positive class is 63.9% of the target variable, we can find a more appropriate threshold by using a portion of the dataset (the calibration set). We thus compute the threshold on the calibration set where we try different t values in $[0.01, 0.99]$, compute the F1 score at each value, and choose the $t^\star$ that maximises binary F1. For LR, this gives $t^\star = 0.56$ (cal F1 = 0.929, against 0.928 at the default $t = 0.5$).

Figure 3 shows the distribution of predicted probabilities $\hat{p}$ on the actual full test set. The decision threshold is at $t^\star = 0.56$ with an example probability band around it: $[t^\star - 0.10,\ t^\star + 0.10] = [0.46, 0.66]$ at deferral rate 10%. If a case whose $\hat{p}$ falls into the band, it will be, by definition, deferred to the SOC analyst. The FPs on the right of the threshold and FNs to the left that fall outside both bands are not deferred, and remain as model errors.

For our case, calibrating the threshold might be not significant, but for heavily imbalanced datasets, the right threshold should be decided based on the probability distribution of the two classes (see Figure 3). Since the RF, XGB and MLP scores are not well-calibrated probabilities, we first map them to probabilities with a Platt (sigmoid) calibration[11]. This gives us $t^\star = 0.56$ for the RF, $t^\star = 0.52$ for the XGBoost, and $t^\star = 0.44$ for the MLP.

[11] `scikit - https://scikit-learn.org/stable/modules/calibration.html`

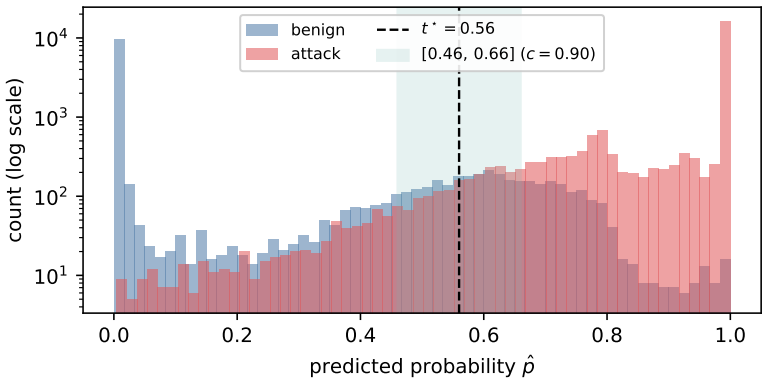


Fig. 3: Distribution of predicted positive ('attack') probabilities $\hat{p}$. The tuned threshold is $t^\star = 0.56$ with a confidence band is around it ($c = 0.90$).

Table 2: Logistic-regression performance metrics.

Metric	Value
Accuracy	0.9093
Precision	0.9111
Recall	0.9509
F1	0.9306
ROC-AUC	0.9772
PR-AUC	0.9863
True neg. (TN)	11,658
False pos. (FP)	2,292
False neg. (FN)	1,212
True pos. (TP)	23,489

Table 3: Deferral-Aware-F1 on the full test set for the four classifiers, each at its own threshold. The last row reports FC-AUC$_{k=0.5}$. Bold marks the best model at each coverage.

Cov. c	Defer.	LR	RF	XGBoost	MLP
1.0	0%	0.9306	**0.9614**	0.9595	0.9517
0.9	10%	0.9618	**0.9888**	0.9877	0.9810
0.8	20%	0.9865	**0.9959**	0.9899	0.9913
0.7	30%	**0.9960**	0.9959	0.9875	0.9913
0.6	40%	**0.9972**	0.9959	0.9834	0.9913
0.5	50%	0.9961	**0.9996**	0.9747	0.9995
FC-AUC$_{k=0.5}$		0.9815	**0.9922**	0.9839	0.9871

Deferral-Aware-F1. Table 2 shows the performance of the LR model in detecting the positive 'attack' class in the test dataset. The model achieves a binary $F_1 = 0.931$ while misclassifying 1,223 attacks (FNs) and 2,279 benign flows (FPs). Table 3 reports DAF1 at the various coverages for all four models. RF appears to be the best model (0.9614), and its DAF1 improves further. XGB also has a good performance, and its DAF1 score obviously improves as well rising to 0.9899 at $c = 0.8$ (20% deferral); however, it also declines due to a relatively lower number of calibrated scores[12]. The DAF1 metric is meant to shows that, for instance, at 30% deferral, the simplest and most explainable model and the last under binary F1 – LR – ties with RF for the best DAF1. More generally, there is an obvious, but significant improvement across all models from zero to

[12] Appendix C details how the threshold and coverage overlay on model probabilities.

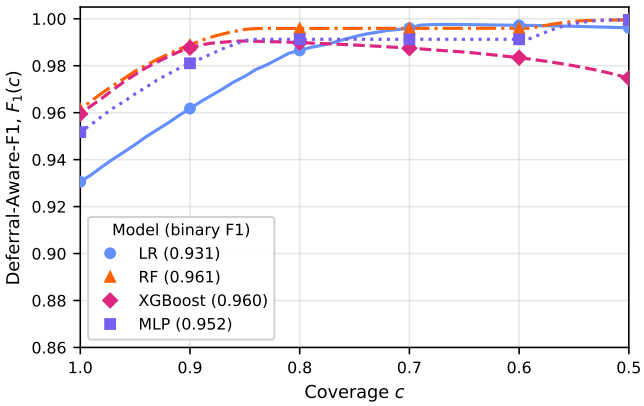


Fig. 4: Deferral-Aware-F1 as a function of coverage c on the full test set for the four classifiers. Coverage decreases (deferral increases) from left to right.

just 5% or 10% deferral rate. Even just with a simple model like LR, a 10% deferral the rate can increase the binary F1 for LR of at least 3%, which means there would be 50% less FPs (-1116) and FNs (-619). All coverage-F1 curves are mostly monotone for F1 above 0.99 or $c = 0.7$.

In a scenario where a SOC manager discusses over different model choices with different stakeholders, they can better inform the audience by showing how a candidate model's performance would score at different human-on-the-loop capabilities or regulation-mandated human supervision requirements.

Of course, it makes little sense to operate above 50% deferral, since most flows would then bypass the automated decision entirely. The DAF1 curve summarizes the trade-off across the whole operating range, but does not by itself pick a specific operating point. The decision-makers should defined what will happen in production. For example, historical data of the SOC case, could help determining realistic deferral rates and expected model performance. The natural next question is how to pick the single operating point that best trades Deferral-Aware-F1 against deferral cost in a way that is easy to translate across different stakeholders. A cost-aware F1 score is meant to capture this need.

4.2 Illustrating Cost-Aware Score

To illustrate the relative cost mapping on the DAF1, we compute the DAF1 with $\alpha \in \{0.0, 0.1, 0.2, 0.3\}$. We thus evaluate $\mathrm{CAS}(\alpha, c)$ at every coverage and find the max $c^\star$ for the three values α for each of the four models.

For simplicity, we first report the CAS metric for the LR in Figure 5 as a set of curves. Every α curve starts with the binary F1 of 0.9306 ($c = 1.0$, no deferral) and climbs into a maximum and then decays (low α, where deferral is rewarded), or slides monotonically downward (high α, where the deferral penalty dominates the F1 gain). Such a representation of model performance aims at visualizing the

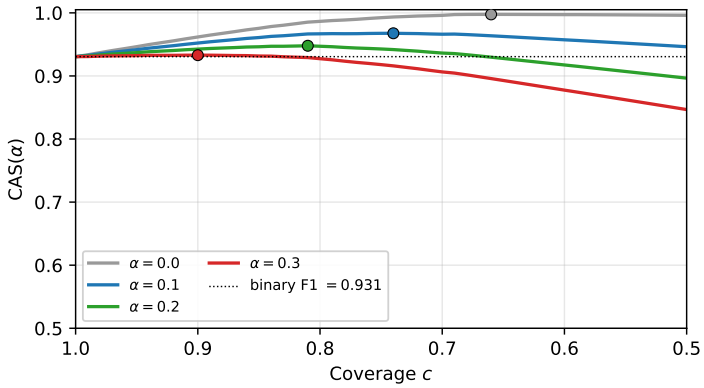


Fig. 5: CAS(α, c) as a function of coverage c on the logistic regression. Each curve is one α value and the point marks the maximum operating coverage $c^\star$.

Table 4: CAS$(\alpha, c^\star)$ optima on the full test set for the four classifiers. For each α we report the optimum coverage $c^\star$, the deferral rate, the model's binary F1 $(c = 1)$, the DAF1 at $c^\star$, its relative gain, the error reduction, CAS(α), and the system-level F1 on the whole test set for an ideal reviewer (F_1^{sys}, Appendix D).

Model	α	$c^\star$	Defer	$F_1(c{=}1)$	$DEF_1(c^\star)$	ΔF_1	$-\Delta$err	CAS(α)	F_1^{sys}
LR	0.1	0.74	26%	0.9306	0.9935	**+6.8%**	**90.6%**	0.9676	**0.9952**
	0.2	0.81	19%	0.9306	0.9853	**+5.9%**	**78.8%**	0.9478	0.9877
	0.3	0.90	10%	0.9306	0.9618	**+3.4%**	**45.0%**	0.9329	0.9647
RF	0.1	0.86	14%	**0.9614**	**0.9946**	+3.5%	86.0%	**0.9807**	0.9950
	0.2	0.91	**9%**	**0.9614**	**0.9871**	+2.7%	66.6%	**0.9693**	0.9879
	0.3	0.95	**5%**	**0.9614**	**0.9777**	+1.7%	42.2%	**0.9631**	0.9785
XGB	0.1	0.89	**11%**	0.9595	0.9889	+3.1%	72.6%	0.9780	0.9899
	0.2	0.91	**9%**	0.9595	0.9860	+2.8%	65.4%	0.9682	0.9870
	0.3	0.95	**5%**	0.9595	0.9762	+1.7%	41.2%	0.9616	0.9772
MLP	0.1	0.85	15%	0.9517	0.9908	+4.1%	81.0%	0.9759	0.9916
	0.2	0.87	13%	0.9517	0.9872	+3.7%	73.5%	0.9616	0.9882
	0.3	0.95	5%	0.9517	0.9680	+1.7%	33.7%	0.9535	0.9691

trade-offs in choosing a given coverage or deferral in a stakeholder conversation about a specific model deployment.

To compare several models, we compute CAS for the LR, RF, XGB and MLP models in Table 4. RF is overall the best performing classifier with the highest binary F1 of 0.96. If we want to include the effect of deferral of the F1 score, we can look at DAF1 which again is best in RF. Notice how LR obtains the highest error reduction rate ($-\Delta$err of 90%), while XGB requires the lowest deferral rate (Defer column). To capture these trade-offs, we can simply look

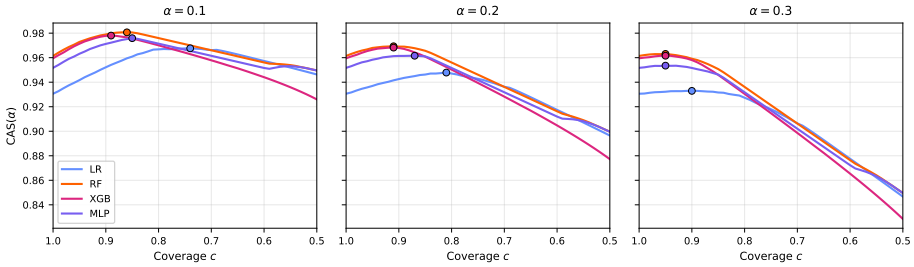


Fig. 6: CAS(α) as a function of coverage c on the test set for the four classifiers. Each dot is a model optimum coverage $c^\star$.

at the CAS value and we see that, for all three α values RF remains the best choice, even if it requires a bit more deferrals than XGB. The pattern is that the CAS optimum coverage increases when the cost of a review α grows: the more expensive analyst time becomes, the lower the deferral rate.

Overall, with $\alpha \leq 0.2$ all four models reach a DAF1 between 0.985 and 0.995 at their CAS optimum, also when the comparison is made on the whole workload rather than on the automated subset (column F_1^{sys}: the F1 of the pipeline including the reviewed cases, for an ideal reviewer). Comparable nominal gains would typically have to come from a heavier detector, with the operational costs that come with it (e.g., GPU inference, raw packet captures instead of aggregated flow records, and less explainable predictions), so that a discussion on what a real-world deployment should look like can steer towards a simpler model with human intervention. In a SOC with many incoming alerts, even small absolute gains in F1 are meaningful (see column $-\Delta$err in Table 4), because each automated FP contributes to alert fatigue and each automated FN is a missed attack. Indeed, the choice of F1 for our metric is dictated by the equal weight placed on FPs and FNs [13].

It is perhaps quicker to appreciate the trade-offs across models and costs in Figure 6 which visualizes the computation of CAS for all four models. Also here it is clear that the RF and XGB ensemble models stay atop: RF is slightly better when reviews become more expensive. Once α is fixed by considering operational considerations, such as analyst hourly cost, average triage time and incident cost of a missed attack, CAS helps deciding on the operating coverage of a SOC. Moreover, by visualizing CAS curves, stakeholders can reason whether deploying a complex and expensive model with lower deferral or a simple classifier with a higher deferral rate.

DAF1 and CAS are computed on the automated subset only, whose composition changes with the coverage, and therefore do not directly measure the performance of the complete human–machine system. Appendix D reports, at the CAS optima of Table 4, the accepted-set precision and recall and a system-

[13] By placing more weight on, say, FNs ($\beta < 1$ in F_β), a deployer can justify the cost of analyst review relative to the potential consequences of missing an attack (FN).

level F1 over the whole workload, for an ideal reviewer and for a reviewer who resolves only a fraction q of the deferred cases correctly.

5 Discussion and Conclusion

The goal of this paper is to showcase that a selective and cost-adjusted F1 score can be used – along with other metrics – to properly evaluate a classifier that will be used in a situation with human supervision. This is needed because many real-world ML deployments have many constraints and requirements, like model complexity, cost, the need to account for the cost of errors, mandatory human reviews and explanability, and many others [12]. Taken together, there is a diverse set of stakeholders who need to navigate across these constraints to decide on the actual system requirements and design.

In Section 4, we showed how the DAF1 and the CAS approaches can account for some of these trade-offs and change the outcome of a model selection. For instance, at 30% deferral the binary-F1 ranking of the four classifiers is reversed for LR and XGB (Table 3), and with $\alpha \leq 0.2$ (medium analyst cost) every model reaches a whole-workload F1 above 0.985 at its CAS optimum (Table 4), a level at which a simple model with human intervention becomes a credible alternative to a more complex and expensive one for a SOC.

Another example of going beyond nominal performance is the DAF1 coverage which allows to give a 'headline number' for a given model performance at a fixed deferral rate. Even better, visualizing the curves (Figure 4 and 5) along a deferral range shows to a wide audience in the boardroom what are the important variables and how do they change. To the engineering stakeholder, the F1 coverage curve tells what the model can do at various points. The operations stakeholder knows what deferral rates are feasible given the required capacity. The business analyst can see the price of a review vs. the cost of a misclassification. More in general, we believe a hybrid approach with models aided by humans can overall help improving the quality of model decisions, event in complex scenarios as LLMs [1].

The same logic, of course, applies to other selective metrics like accuracy, ROC/PR-AUC curves and - the original idea - selective risk [8,10]. In this paper, we chose F1 as it is a 'safe' metric that accounts for a balanced precision and recall in a rare-event setting for many scenarios, like a SOC, but often also in healthcare and financial applications.

Importantly, the DAF1 and CAS approach is meant to visualize a complex trade-off between cost and benefit in practical human-in-the-loop deployments, and illustrate the real benefit of employing a specific model for diverse stakeholders in a more intuitive and realistic way. They are not meant to replace the metrics above or the existing learning with rejection tools and methods, like rejection estimators [22], which remain the right tools for model-with-rejection developers [16]. Any model with a trained rejection estimator should be used instead of our confidence-based mechanism. DAF1 and CAS come into play only when a trained classifier (without rejection) needs to be evaluated for deploy-

ment with a huamn review budget and where the trade-off has to be explained and agreed upon by people who may be not familiar with risk-coverage curves.

Limitations. One important limitation of this paper is that evaluates new metrics in only one scenario and one dataset. It would be interesting to understand the effect when using different dataset. Another limitation is that we use existing metrics like F1 that have their own limitations, such as ignoring a potential difference in cost between false positives and false negatives. This means that deployments that have a different balance of costs of errors, might need a different coverage-aware metric (e.g. risk [10]).

CAS assumes that every review has the same cost, making this communicable across stakeholders, but it is also a strong simplification. Moreover, we assume an ideal reviewer who resolves every deferred case correctly. In reality, with imperfect reviewer, the deferral gain decreases making this a more complex problem as studied in the learning-to-defer literature [20,26]. Appendix D quantifies this with a system-level F1 for an imperfect reviewer correctly resolving a fraction of cases.

Another important limitation is that our deferral rule ranks cases by confidence $|\hat{p} - t^{\star}|$. This works well when a model's errors concentrate near its decision boundary (i.e. our threshold, see Figure 3). Therefore this makes our metrics effectiveness dependent on how close to the threshold(s) are the errors (confident but wrong predictions are never captured) and how the cases are distributed across the classes (one threshold does not work well with strong class imbalance). One can work around it by designing the rejection rule based on the probabilities/ranks distribution as done in classification with rejection literature [16] or with two thresholds proportional to the distributions of the two classes in binary classification (e.g, we exemplify a simple 2-threshold approach in Appendix C). Similarly, the deferral rule can be based on outlier detection whereby any input with an outlier score higher than a specified threshold would be deferred to a human reviewer.

Future Work. In the future, we would like to test this metric in real world applications, and see if this metric indeed leads to more trust in the use of machine learning in real business decisions or fewer objections to decisions. We hope we can validate the metrics with a user study in real requirement discussions with real decision-makers. Another direction for future research is to extend the use of our metrics even for classifiers that do not provide usable confidence scores. Also, a two-sided approach for a future development of DAF1 will keep the class-specific coverage constant as the coverage changes, thus making it independent from the varying prevalence among the classes.

Acknowledgements We would like to thank Yiannis Kanellopoulos for useful discussions and the reviewers for their valuable feedback.

References

1. Abels, A., Lenaerts, T.: Wisdom from Diversity: Bias Mitigation Through Hybrid Human-LLM Crowds (Oct 2025), https://openreview.net/forum?id=SgwOpalNQo
2. Alahmadi, B.A., Axon, L., Martinovic, I.: 99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. pp. 2783–2800 (2022), https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi
3. Angius, R.: Stavolta Piracy Shield l'ha fatta grossa e ha bloccato Google Drive (Oct 2024), https://www.wired.it/article/piracy-shield-blocco-google-drive-download/, section: tags
4. Bank, D.N.: Perspectives on Explainable AI in The Financial Sector. Tech. rep., De Nederlandsche Bank (2021)
5. Bank, D.N.: AI bij verzekeraars: kansen en risico's (Mar 2025), https://www.dnb.nl/nieuws-voor-de-sector/toezicht-2025/ai-bij-verzekeraars-kansen-en-risico-s/
6. Board, E.D.P.: Automated decision-making and profiling (2018), https://www.edpb.europa.eu/documents/guideline/automated-decision-making-and-profiling_en
7. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0: Step-by-step data mining guide (2000), https://www.semanticscholar.org/paper/CRISP-DM-1.0%3A-Step-by-step-data-mining-guide-Chapman/54bad20bbc7938991bf34f86dde0babfbd2d5a72
8. Chow, C.K.: On Optimum Recognition Error and Reject Tradeoff. IEEE Transactions on Information Theory **16**(1), 41–46 (1970). https://doi.org/10.1109/TIT.1970.1054406
9. Condessa, F., Bioucas-Dias, J., Kovačević, J.: Performance measures for classification systems with rejection. Pattern Recognition **63**, 437–450 (2017). https://doi.org/10.1016/j.patcog.2016.10.011
10. El-Yaniv, R., Wiener, Y.: On the Foundations of Noise-Free Selective Classification. Journal of Machine Learning Research **11**, 1605–1641 (2010)
11. Fischer, L., Wollstadt, P.: Precision and recall reject curves for classification. arXiv:2308.08381 (2023), https://arxiv.org/abs/2308.08381
12. García, L.A., Loeckx, J.: Design Matters: Unveiling the Impact of AI Engineering Decisions on Alignment - A Case Study in School Dropout Prediction (Oct 2025), https://openreview.net/forum?id=Li1ND2J8hW
13. Geifman, Y., El-Yaniv, R.: Selective Classification for Deep Neural Networks (Jun 2017). https://doi.org/10.48550/arXiv.1705.08500, arXiv:1705.08500 [cs.LG]
14. Hanczar, B.: Performance visualization spaces for classification with rejection option. Pattern Recognition **96**, 106984 (2019). https://doi.org/10.1016/j.patcog.2019.106984
15. Hasler, S., Fischer, L.: Stacked confusion reject plots (SCORE). arXiv:2406.17346 (2024), https://arxiv.org/abs/2406.17346
16. Hendrickx, K., Perini, L., Plas, D.V.d., Meert, W., Davis, J.: Machine Learning with a Reject Option: A survey (Feb 2024). https://doi.org/10.48550/arXiv.2107.11277, arXiv:2107.11277 [cs.LG]
17. Herbei, R., Wegkamp, M.H.: Classification with Reject Option. The Canadian Journal of Statistics **34**(4), 709–721 (2006)

18. Kersten, L., Mulders, T., Zambon, E., Snijders, C., Allodi, L.: 'Give Me Structure': Synthesis and Evaluation of a (Network) Threat Analysis Process Supporting Tier 1 Investigations in a Security Operation Center. pp. 97–111 (2023), https://www.usenix.org/conference/soups2023/presentation/kersten
19. Leijnen, S., Maathuis, H., van Montfort, K., Van Otterloo, S., Sent, D., Stalenhoef, M., van Turnhout, K., Zwaal, R.: Guideline for user interface design of explainable AI. Tech. rep., Hogeschool Utrecht (2025), https://www.hu.nl/onderzoek/publicaties/guideline-for-user-interface-design-of-explainable-ai
20. Madras, D., Pitassi, T., Zemel, R.: Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper_files/paper/2018/hash/09d37c08f7b129e96277388757530c72-Abstract.html
21. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) (Nov 2015). https://doi.org/10.1109/MilCIS.2015.7348942
22. Mozannar, H., Sontag, D.: Consistent Estimators for Learning to Defer to an Expert (Jan 2021). https://doi.org/10.48550/arXiv.2006.01862, arXiv:2006.01862 [cs.LG]
23. Parliament, E.: Art. 22 GDPR – Automated individual decision-making, including profiling (2018), https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng
24. Parliament, E.: Article 14: Human Oversight | EU Artificial Intelligence Act (2024), https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02024R1689-20260727
25. Persoonsgegevens, A.: Meaningful human intervention in algorithmic decision-making (Jul 2025), https://www.autoriteitpersoonsgegevens.nl/en/current/meaningful-human-intervention-in-algorithmic-decision-making
26. Strong, J., Sun, E., Rogers, H., Higham, H., Noble, A.: Learning to Defer: A Survey (Dec 2025). https://doi.org/10.5281/zenodo.17843044

A Dataset & Explainability

A.1 Dataset

Tables 5 and 6 show the summary of the UNSW-NB15 dataset after preprocessing and its features with descriptions.

Table 5: UNSW-NB15 dataset summary after preprocessing.

Property	**Value**
Raw rows / columns	257,673 / 45
Dropped columns	2 (`id`, `attack_cat`)
One-hot encoded categoricals	`proto`, `service`, `state`
Usable features after encoding	196
Target variable	`label` (binary)
Positive class prevalence	63.91%
Train / calibration / test split	70/15/15 stratified
Train / cal / test sizes	180,371/38,651/38,651

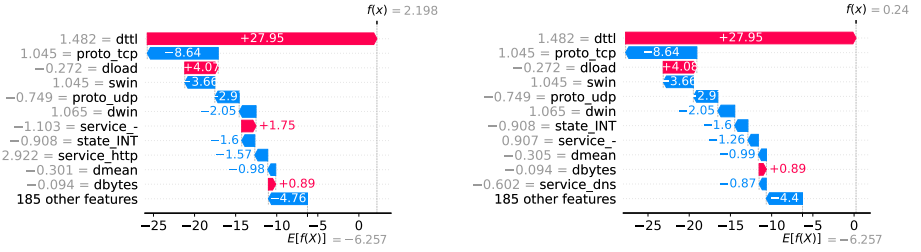
A.2 Explainability

Simple models are often preferred because they allow for local explanations of a given model decision, which is often required by law or practice [5,25,19]. Logistic regression is a good example that can explain a decision by giving weights for each factor contributing to the outcome. Figure 7 shows SHAP waterfall plots with the relative weights of the most important factors for an 'attack' label, for two correctly detected attacks: a confident one (Figure 7a) and one close to the confidence threshold (Figure 7b). The brighter colour pushes the decision towards 'attack', the darker towards 'benign'. Both decisions are heavily conditioned on the destination TTL (`dttl`, +27.95) and share ten of the twelve displayed contributions: the ≈2 log-odds that separate the confident detection (a) from the borderline one (b) come almost entirely from the `service` feature. This allows the analyst to reason on what can make this prediction a not confident positive (attack) class and whether the decisive factor (`service`) actually makes sense in the operation al domain. The magnitude of the attributions therefore carries no information about confidence, only the final value $f(x)$ distinguishes the two cases (significantly above zero in Figure 7a).

This would not be possible for more black-box models like complex neural networks. Note, however, the local attribution tells the analyst which features condition the score, but not whether the resulting decision is reliable enough to act on.

Table 6: UNSW-NB15 feature schema, grouped by feature group. The `attack_cat` column names one of nine attack types or `Normal` for benign traffic.

Group	Columns	Meaning
Flow basics	`dur`, `proto`, `service`, `state`, `spkts`, `dpkts`, `sbytes`, `dbytes`, `rate`	Connection duration, protocol, byte/packet counts
Network layer	`sttl`, `dttl`, `sloss`, `dloss`, `sload`, `dload`, `swin`, `dwin`, `stcpb`, `dtcpb`	TTL, packet loss, load, TCP window/sequence
Derived/ statistical	`smean`, `dmean`, `sinpkt`, `dinpkt`, `sjit`, `djit`, `tcprtt`, `synack`, `ackdat`, `trans_depth`, `response_body_len`	Packet sizes, jitter, RTT, HTTP depth
Connection-table	`ct_srv_src`, `ct_state_ttl`, `ct_dst_ltm`, ..., `ct_dst_src_ltm`	Windowed connection counts (last 100 connections)
Binary flags	`is_ftp_login`, `is_sm_ips_ports`	FTP login indicator, same-IP-and-port indicator
Target	`label`	Binary label (0=normal, 1=attack); `attack_cat` is ignored



(a) High-confidence attack ($p = 0.90$, $f(x) = +2.20$).

(b) Low-confidence attack ($p = 0.56 = t^*$, $f(x) = +0.24$).

Fig. 7: SHAP explanations for two correctly detected attacks. Each bar is a feature contribution to the model's log-odds output, accumulating from the dataset average baseline (all features average, $E[f(X)]$=-6.26) to the specific attack score $f(x)$. Feature values are standardised

B F1-Coverage AUC (FC-AUC)

The coverage-F1 curve in Section 2.1 can be summarized in a single number by computing the Area Under Curve (AUC) in the same fashion as AURC [13]. Substituting F1 for error rate in that construction gives us the *F1-Coverage AUC* (FC-AUC). We compute a simple average on curve with the trapezoidal rule for simplicity: let $c_1, c_2, \ldots, c_M$ be M coverage values evenly spaced over

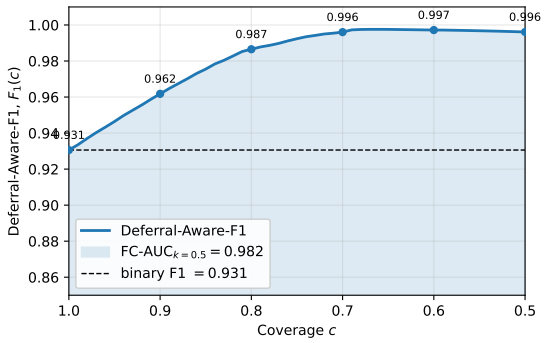


Fig. 8: Deferral-Aware-F1 curve of the logistic regression on the test set at threshold $t^\star = 0.56$ with the FC-AUC shaded area.

$[k, 1]$ with $c_1 = k$ and $c_M = 1$. We define FC-AUC$_k = \frac{1}{M}\sum_{i=1}^{M} F_1(c_i)$, where $F_1(c_i)$ is the selective F1 evaluated on the most-confident $c_i N$ test cases under the rejection rule of Section 2.1. Each $F_1(c_i)$ is between 0 and 1, therefore also the FC-AUC is between 0 and 1. We use $k = 0.5$ as the default because no real classifier deployment (including a SOC) would defer more than half of model decisions.

Like AURC, FC-AUC summarises the curve over a range of operationally plausible deferral rates. It can be a useful "one-number metric" for stakeholders deciding on a deployment, especially when a simple (cheaper) model paired with a human reviewer might be a better alternative to a more complex and expensive model. This is particularly relevant in sensitive decision-making tasks where regulatory oversight mandates reviews anyway.

Figure 8 shows FC-AUC on the Deferral-Aware-F1 curve of the logistic regression illustration from our the Logistic Regression illustration Section 4. The shaded area corresponds to the FC-AUC$_{k=0.5} = 0.9815$.

C Deferral under strong imbalance

The dataset we use in the main text (Section 3) is rebalanced by the dataset authors on Kaggle to 63.91% attack prevalence. The underlying raw dataset from the original paper UNSW-NB15 [21] is more imbalanced: the raw files (`UNSW-NB15_1.csv`–`UNSW-NB15_4.csv`, 2,540,047 connections) have an attack rate of 12.65%, about 5 times more than the Kaggle dataset.

As noted in the limitations (Sec. 5), our deferral argument is dependent on how the residual model uncertainty is distributed (closer to the threshold, the better). Since imbalance shifts this uncertainty, we repeat the logistic regression illustration of Section 4 on the raw dataset. We perform stratified sampling of 250,000 connections preserving the 12% prevalence. Preprocessing follows Section 3.2 with any extra feature not in Sectio 3.2 dropped.

Table 7: Selective F1 of the LR on raw network flow in the original UNSW-NB15 dataset (12.65% attack prevalence, threshold $t^\star = 0.90$), with one-sided and two-sided coverage. "pos. kept" is the share of all test-set attacks (positive class) still present in the accepted subset. Under one-sided coverage the accepted set has not more attacks almost immediately, so F1 goes to zero. Two-sided coverage keeps the positives in proportion to c ($\mathcal{A}_1(c)$) and the curve follows the behavior as in Section 4.

		one-sided		**two-sided**	
c	deferral	F1	pos. kept	F1	pos. kept
1.0	0%	0.9585	100.0%	0.9585	100.0%
0.9	10%	0.9818	28.9%	0.9958	89.3%
0.8	20%	0.0000	0.0%	0.9995	79.9%
0.7	30%	0.0000	0.0%	0.9994	69.9%
0.6	40%	0.0000	0.0%	0.9993	59.9%
0.5	50%	0.0000	0.0%	0.9992	49.9%

One-sided coverage. Applying the coverage rule of Section 2.1, Deferral-Aware-F1 does rise from 0.9592 to 0.9806 at $c = 0.9$, but at $c = 0.8$ collapes to zero because of how the probability scores are distributed: Figure 9a) show the distribution where the skewness toward small confidence score for attack cases is clear. With a threshold of $t^\star = 0.90$, the mean confidence for is 0.89 for benign cases and just 0.093 for an attack case. At some point, by lowering coverage less than 0.87, we begin discarding real attack cases instead of errors.

For this reason, we chose to use the balanced dataset to simplify the explanation of the core concepts instead of going in the details of model scores' distribution. To solve this problem, we can use rank the two predicted classes separately.

Two-sided coverage. Instead of ranking the test set as a whole, we can partion it by predicted class, $A_1 = \{i : \hat{p}_i \geq t\}$ (points predicted per attack) and $A_0 = \{i : \hat{p}_i < t\}$ (points predicted per not-attack), and take the most confident c fraction within each partition: the set of top accepted decisions per class are $\mathcal{A}_k(c) =$ the $\lceil c|A_k| \rceil$ points of A_k with the largest $|\hat{p}_i - t|$ with $k = 1$ for positive class and $k = 0$ for negative class.

$\mathcal{A}_k(c)$ is the set accepted model decisions at coverage c. The accepted subset then preserves the predicted-class ratio at every coverage, so F1 remains well defined. In our concrete example of SOC, the interpretation is does not change: if we defer 20% of its volume defers 20% of its uncertain *attacks* and 20% of its *no-attack* alerts. Therefore, the share of retained (not deferred) attacks follows c almost exactly (89.4%, 80.1%, 70.0%, 60.0%, 50.0% at $c = 0.9, \ldots, 0.5$; Figure 9b), and Deferral-Aware-F1 has the monotone curve seen in Section 4: from 0.9592 at $c = 1$ to 0.9995 at $c = 0.8$ (Table 7, right side).

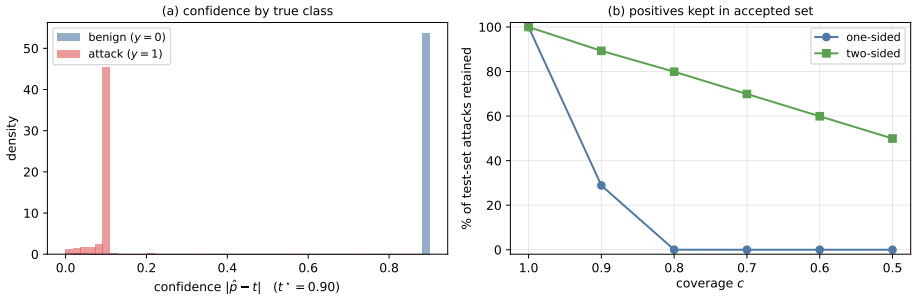


Fig. 9: (a) Confidence $|\hat{p} - t|$ by true class: benign traffic is furthest from the threshold and is therefore ranked most confident. (b) Part of test-set attacks retained in the accepted subset as coverage falls.

Table 8: CAS(α) on the raw-capture test set ($t^\star = 0.90$, binary F1 $= 0.9592$). Both coverage definitions place the optimum at high coverage and agree to within 0.01.

	one-sided			two-sided		
α	$c^\star$	$F_1(c^\star)$	CAS	$c^\star$	$F_1(c^\star)$	CAS
0.1	0.97	0.9927	0.9897	0.92	0.9941	0.9862
0.2	0.97	0.9927	0.9867	0.94	0.9913	0.9794
0.3	0.97	0.9927	0.9837	0.98	0.9824	0.9765

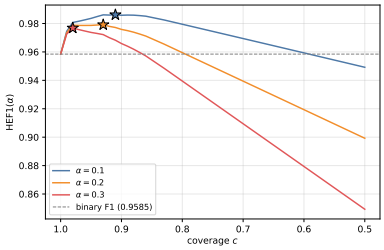


Fig. 10: CAS(α) against coverage (two-sided); stars mark the per-α optimum, dashed line the binary F1.

We can thus appreciate the effect of two sided ranking also on CAS: Table 8 reports CAS(α) under both definitions: the optimal coverage sits at $c^\star \geq 0.92$, and the CAS values agree to within 0.01; Figure 10 shows the two-sided case for CAS curves.

D Accepted-set metrics and system-level F1

DAF1 evaluates the accepted subset $\mathcal{A}(c)$ only. As the coverage decreases, the least confident, and likely hardest, cases are removed from this subset and its class mix changes (see the limitations in Section 5), so an increase in DAF1 is not by itself an improvement of the complete pipeline. Table 9 therefore reports, at the CAS optima $c^\star(\alpha)$ of Table 4, (1) the precision and recall on $\mathcal{A}(c)$, which show how the F1 improvement is differs between fewer false positives and fewer false negatives, and (2) a *system-level* F1 that scores the complete human–machine pipeline on all N test cases and not only on the automated subset.

Let $\mathrm{TP}_{\mathcal{A}}$, $\mathrm{FP}_{\mathcal{A}}$ and $\mathrm{FN}_{\mathcal{A}}$ be the counts on the accepted subset and $P_{\mathcal{D}}$ and $N_{\mathcal{D}}$ the numbers of actual positives and negatives among the deferred cases. If the human reviewer classifies each deferred case correctly with probability q, independently of its class, the expected counts of the pipeline are $\mathrm{TP}_{\mathrm{sys}} = \mathrm{TP}_{\mathcal{A}} + q\,P_{\mathcal{D}}$, $\mathrm{FN}_{\mathrm{sys}} = \mathrm{FN}_{\mathcal{A}} + (1-q)\,P_{\mathcal{D}}$ and $\mathrm{FP}_{\mathrm{sys}} = \mathrm{FP}_{\mathcal{A}} + (1-q)\,N_{\mathcal{D}}$, so that

$$F_1^{\mathrm{sys}}(c,q) \;=\; \frac{2\,\mathrm{TP}_{\mathrm{sys}}}{2\,\mathrm{TP}_{\mathrm{sys}} + \mathrm{FP}_{\mathrm{sys}} + \mathrm{FN}_{\mathrm{sys}}}$$

For the ideal reviewer assumed in Section 2 ($q = 1$) the only errors left in the pipeline are those of the accepted subset, and $F_1^{\mathrm{sys}}(c,1)$ is the F1 that the SOC would observe on its entire workload. For $q < 1$ the deferred cases contribute errors of their own and $F_1^{\mathrm{sys}}(c,q)$ has an interior optimum roughly where the reviewer's error rate $1-q$ exceeds the model's error rate on the cases being deferred, which connects our metrics to the learning-to-defer setting [20,22].

Table 9: Accepted-set precision and recall and system-level F1 at the CAS optima $c^\star(\alpha)$ of Table 4.

Model	α	$c^\star$	Defer	DAF1	CAS	$\mathrm{Prec}_{\mathcal{A}}$	$\mathrm{Rec}_{\mathcal{A}}$	$F_1^{\mathrm{sys}}(c,q)$		
								q=1	q=0.9	q=0.8
LR	0.1	0.74	26%	0.9935	0.9677	0.9959	0.9911	**0.9952**	0.9747	0.9539
	0.2	0.81	19%	0.9852	0.9478	0.9844	0.9861	0.9877	0.9728	0.9578
	0.3	0.90	10%	0.9618	0.9329	0.9502	0.9736	0.9647	0.9570	0.9492
RF	0.1	0.86	14%	0.9946	0.9807	0.9907	0.9985	0.9950	**0.9842**	**0.9733**
	0.2	0.91	9%	0.9871	0.9693	0.9855	0.9887	0.9879	0.9809	0.9739
	0.3	0.95	5%	0.9777	0.9630	0.9782	0.9772	0.9785	0.9746	0.9707
XGB	0.1	0.89	11%	0.9889	0.9780	0.9962	0.9816	0.9899	0.9813	0.9726
	0.2	0.91	9%	0.9860	0.9682	0.9920	0.9800	0.9870	0.9799	0.9728
	0.3	0.95	5%	0.9762	0.9616	0.9822	0.9703	0.9772	0.9732	0.9693
MLP	0.1	0.85	15%	0.9908	0.9759	0.9818	0.9999	0.9916	0.9800	0.9685
	0.2	0.87	13%	0.9872	0.9616	0.9778	0.9969	0.9882	0.9782	0.9682
	0.3	0.95	5%	0.9680	0.9535	0.9643	0.9718	0.9691	0.9652	0.9614

With a human reviewer who classifies 90% of the deferred cases correctly, the system-level F1 at every CAS optimum of Table 9 still exceeds the binary F1 of the same model; at 80% it still does at the optima, although in our experiments deferring 30% or more of the cases is no longer convenient for RF, XGB and MLP (not shown), i.e., the review budget suggested by CAS is also the range in which an imperfect reviewer still adds value.