
FULL-RANGE BINARY CLASSIFIER CALIBRATION FOR STABLE MODEL UPDATES IN PRODUCTION

Konstantin Berlin
Cisco AI Defense

Abstract

Detection models running in adversarial environments face a malicious distribution that drifts rapidly while the benign distribution stays comparatively stable, so teams retrain and redeploy constantly to stay ahead of new threats. Retraining tends to change the output prediction scores, which breaks downstream users of the model. For these security-oriented models we need consistent false-positive rate (FPR) across all output values, whereas standard probability-calibration methods target class probability rather than an FPR contract. We introduce a method built on top of existing calibration primitives that targets the whole FPR curve, giving scores a consistent FPR meaning across deployments. On one held-out split, the observed relative FPR error was at most 2.3% from 10% down to 0.1% FPR and 7.2% at 0.01% FPR. The shipped artifact remains under 200 KB in measurements across calibration sets from 1K to 10M benign samples.

1 Introduction

Detection models running in adversarial environments face a malicious distribution that drifts rapidly while the benign distribution stays comparatively stable. Teams therefore retrain and redeploy detection models continuously to stay ahead of evolving threats [1]. Retraining tends to change the output prediction scores, which breaks downstream users of the model. For these security-oriented models we need consistent false-positive performance across all output values. Standard probability-calibration methods such as Platt scaling and isotonic regression calibrate class probability rather than an FPR contract [2, 3]. We introduce a method that calibrates raw scores to the false-positive rate (FPR) directly on benign samples only, composing existing sklearn primitives (`MinMaxScaler` and `IsotonicRegression`) into a `Pipeline` that requires no custom inference code. The two-spline construction in Figure 1 keeps the FPR-to-rescaled-score lookup at fit time and ships only a raw-score-to-calibrated-score pipeline for production inference. Here, stability means that each model release is calibrated independently to the same fixed FPR-to-score contract. Raw-score thresholds may change after retraining, but a calibrated threshold retains the same target benign FPR.

The FPR target depends only on the benign distribution. In the update setting we target, calibrating to FPR uses only benign traffic, which is the larger and more directly measurable sample pool, and avoids having to characterize adversarial behavior. In adversarial detection, positive examples are unknown unknowns because new attacks are hard to enumerate and label, yet their count remains far smaller than the benign count. FPR calibration is robust to positive-count uncertainty because it divides by the large, well-characterized benign count, whereas precision divides by the small, poorly characterized predicted-positive count, as quantified by the binomial planning rule in Section 2.

A single calibrated score can feed multiple downstream tiers (e.g., block at 0.1% FPR, alert at 1%, escalate at 10%), so calibration must hold across thresholds rather than only at one operating point.

The method’s contributions are:

1. **Whole-curve FPR mapping via a non-parametric monotone linear spline.** Isotonic regression over the benign empirical CDF maps score thresholds to FPR at every threshold, with no assumed functional form.

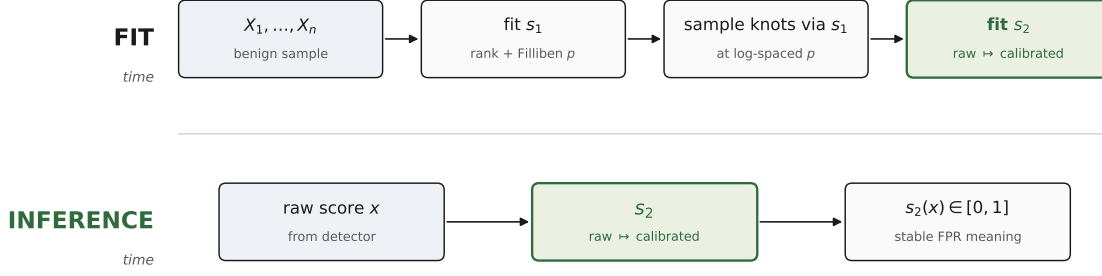


Figure 1: Method overview. During fitting, the calibrator sorts benign scores, assigns plotting-position FPR labels, and fits a temporary FPR-to-rescaled-score spline. A fixed log-spaced FPR base grid augmented with that spline’s fitted edge knots then passes through the spline and the log-scale output contract to produce rescaled-score-to-calibrated-score storage knots. The second spline is the shipped sklearn artifact, so inference uses only raw score, fixed scaling, and the rescaled-score-to-calibrated-score map.

2. **Fixed log-scale output contract.** A piecewise-linear map in $\log_{10}(\text{FPR})$ space pins calibrated score 0.5 to 0.1% FPR, 0.7 to 0.01%, and 0.85 to 0.001%. Each step between anchors is a tenfold change in rarity, so the calibrated axis reads uniformly to an operator.
3. **Knot-subsampled, fixed-size deployable artifact.** A second isotonic fit on a fixed log-spaced FPR base grid augmented with fitted edge knots produces a calibration artifact under 200 KB in our measurements, independent of calibration-set size.
4. **Below-floor extrapolation with safe clipping.** Linear extrapolation from the two lowest observed (FPR, score) points, composed through the log anchors, keeps the output monotone and bounded below the sample-supported FPR floor.
5. **Release-specific calibration under a fixed cross-retraining contract.** Each model release refits its calibration artifact on that model’s benign scores while retaining the fixed FPR-to-score anchors, so downstream systems can keep the same calibrated thresholds across releases.

Section 5 reports anchor-level calibration error and full-curve diagnostics on the held-out-from-fit subset, with the calibration-fit subset shown separately.

2 FPR estimate bounds and biases

Two finite-sample effects bound FPR estimation accuracy for any calibrator: sampling variance on the benign set and edge-of-sample bias when rank-selected score thresholds receive FPR labels for spline knots.

2.1 Sampling variance

Let S_i be the model’s scalar prediction score on benign calibration input i for $i = 1, \dots, n$, and let S_{benign} be the score the same model would assign to a fresh benign input from the same distribution. For a fixed score threshold τ , each benign calibration example either fires or does not fire, so the empirical FPR is a proportion over n independent 0/1 outcomes:

$$\hat{q}(\tau) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{S_i \geq \tau\}. \quad (1)$$

The standard normal approximation for a binomial proportion gives the 95% interval [4]

$$\hat{q}(\tau) \pm 2\sqrt{\frac{\hat{q}(\tau)(1 - \hat{q}(\tau))}{n}}. \quad (2)$$

For target FPR p and relative half-width r , converting Equation (2) gives the planning rule derived in Section A:

$$n \approx \frac{4}{r^2 p}. \quad (3)$$

Table 1 reports Equation (3) for common deployment FPRs. At a target FPR of 0.1%, the normal approximation recommends 64,000 benign samples, corresponding to a rough 95% range of 0.075%–0.125%.

Table 1: Benign sample sizes estimated using the normal approximation.

Target FPR p	$r = 50\%$	$r = 25\%$	$r = 10\%$
10^{-1}	160	640	4,000
10^{-2}	1,600	6,400	40,000
10^{-3}	16,000	64,000	400,000
10^{-4}	160,000	640,000	4.0M
10^{-5}	1.6M	6.4M	40M
10^{-6}	16M	64M	400M

2.2 Edge-of-sample bias

At a fixed score threshold, Section 2.1 measures counting error. The first calibration spline has a different problem because it starts from sorted benign scores and must attach one FPR label to each rank-selected score threshold. For a fixed threshold τ , the fresh-sample FPR is

$$q(\tau) = \Pr(S_{\text{benign}} \geq \tau), \quad (4)$$

where S_{benign} is the model score of a fresh benign deployment example.

Let $k \in \{1, \dots, n\}$ count ranks from the largest benign model score downward, and let τ_k be the k -th largest benign model score. The fresh-sample FPR at that rank-selected threshold is

$$q_k = q(\tau_k) = \Pr(S_{\text{benign}} \geq \tau_k). \quad (5)$$

The spline label attached to τ_k is written $\tilde{q}_k$ to distinguish it from the empirical FPR estimate $\hat{q}(\tau)$ in Equation (1). The naive spline label is

$$\tilde{q}_k^{\text{sample}} = \frac{k}{n}. \quad (6)$$

Equation (6) is an in-sample count: k of the n calibration scores are at or above τ_k . The spline needs the fresh-sample value q_k , the probability that a fresh benign score exceeds the threshold selected by that rank. Because τ_k was chosen after sorting, it already had to beat $n - k$ calibration scores, so a fresh benign score exceeds it less often than k/n suggests. With n sorted samples creating $n + 1$ probability gaps, the k -th largest score has mean fresh-sample FPR [5]

$$\tilde{q}_k^{\text{mean}} = \frac{k}{n + 1}. \quad (7)$$

Since $k/(n + 1) < k/n$, the naive knot assigns too large an FPR label to τ_k and returns a threshold that is slightly more selective than requested.

The mean correction shows that the naive label k/n is biased high for the fresh-sample FPR. The implementation uses Filliben because the spline must choose one typical label per rank, and Filliben is the standard median plotting-position rule. Filliben uses order-statistic medians rather than means for probability-plot locations [6], and the NIST probability-plot formula gives the interior uniform median approximation [7]. After flipping from lower-tail percentile to upper-tail FPR, the interior-rank label is

$$\tilde{q}_k^{\text{Filliben}} = \frac{k - 0.3175}{n + 0.365}. \quad (8)$$

The implementation uses Filliben by default, with endpoint values $1 - 0.5^{1/n}$ for the largest benign score and $0.5^{1/n}$ for the smallest. Held-out evaluation still reports count-based FPR; Filliben is used only to place first-spline knots during fitting.

Filliben reduces relative label bias, not relative sampling noise. At $p = 10^{-3}$ with 50,000 benign calibration samples, the relevant tail rank is about 50. Filliben moves the naive label by about 0.6% relative error, while the sampling interval from Equation (2) is about $\pm 28\%$ relative error. The correction is justified, but the practical error is dominated by having only about 50 benign tail examples. Figure B.1 shows that Filliben removes the median relative label error, and Figure B.2 shows that mean absolute relative label error barely changes because rank-selected thresholds still move across calibration sets. Only more benign calibration samples materially reduce that dominant relative error, as Table 1 shows.

3 Method

The calibrator builds a shipped raw-score-to-calibrated-score pipeline from a temporary FPR-to-rescaled-score spline and a fixed FPR-to-calibrated-score contract. Figure 1 shows the fit-time objects and the shipped object. We use `IsotonicRegression` because it is sklearn’s built-in monotone piecewise-linear spline and fits directly inside an sklearn `Pipeline`. In this method, it serves mainly as a deployable spline container. The implementation will be released at <https://github.com/cisco-ai-defense/fpr-model-calibration>.

Log-scale output anchors. Log spacing matches deployment practice because operators perceive FPR changes by order of magnitude rather than by linear differences in probability. The calibrated axis therefore starts from a fixed contract in $\log_{10}(\text{FPR})$ space before any detector-specific spline is fit. Table 2 defines the full anchor map, including the convention that a calibrated score of 0.5 corresponds to 0.1% FPR. Pinning these values gives the calibrated score the same alert-rate meaning across model versions and detector categories. The pipeline emits calibrated scores no greater than 0.99, so a threshold of 1.0 flags nothing.

Table 2: Log-scale anchor map from FPR to calibrated score.

FPR	Calibrated threshold	Interpretation
100%	0.00	Flag everything
10%	0.10	1 in 10 benign flagged
1%	0.30	1 in 100
0.1%	0.50	1 in 1,000
0.01%	0.70	1 in 10,000
0.001%	0.85	1 in 100,000
0.0001%	0.95	1 in 1,000,000
0%	1.00	Flag nothing

Scale scores. A `MinMaxScaler` with `feature_range=(0, 0.99)` fit on the input domain $[0, 1]$ maps raw scores to $[0, 0.99]$. Using the fixed input domain keeps production scores above the calibration-time maximum inside the trained range.

Spline 1: FPR to rescaled score. The first spline answers the fit-time lookup from target FPR p to the rescaled score threshold that should carry it. The finite benign sample supplies thresholds only at its rank labels, so most desired FPR values on the log grid have no exact observed score. This temporary spline interpolates those missing thresholds during fitting and is not shipped. Sort the n rescaled benign scores as $s_{(1)} \leq \dots \leq s_{(n)}$. For score $s_{(j)}$, let $k = n - j + 1$ be its rank from the largest score downward. Attach the Filliben FPR label $\hat{q}_k^{\text{Filliben}}$ from Equation (8) to $s_{(j)}$. Fit `IsotonicRegression(increasing=False)` on these (FPR label, rescaled score) pairs.

Synthetic knot grid. Start from a fixed log-spaced base grid of $\sim 10,000$ FPR values, with explicit grid points at every decade from 10^{-10} to 1. Augment the base grid with Spline 1’s high-FPR endpoint and the two lowest-FPR knots that define extrapolation. For each resulting FPR, query the FPR-to-rescaled-score spline to get a rescaled-score threshold and query the fixed log-scale FPR-to-calibrated-score contract to get the calibrated score. For FPRs below Spline 1’s smallest observed FPR label, replace isotonic clipping with linear extrapolation from the two lowest observed (FPR, rescaled score) knots, bounded inside $[0, 0.99]$. These generated (rescaled score, calibrated score) pairs are storage knots for the shipped spline, not new statistical evidence below the sample-size floor.

Spline 2: rescaled score to calibrated score. Fit `IsotonicRegression(increasing=True)` on the generated (rescaled score, calibrated score) pairs. Include boundary training pairs (0.99, 0.99) and (1.0, 1.0) in the rescaled-score-to-calibrated-score fit. If tied benign scores produce repeated score knots, `IsotonicRegression` averages the tied block while preserving monotonicity. The shipped artifact is an sklearn `Pipeline` with the `MinMaxScaler` followed by the second `IsotonicRegression`, so production inference uses standard sklearn objects rather than custom spline code.

4 Fixed-size deployable artifact

Production deployment needs artifact storage bounded independently of calibration-set size. Let K denote the number of breakpoints retained by the fitted `IsotonicRegression`. An uncapped fit stores up to one

breakpoint per distinct training score, so K can approach n . Its two `float64` breakpoint arrays require $16K$ bytes; at $K = 10$ million, those arrays alone require 160 MB.

The knot-subsampling step in Section 3 caps K at approximately 10,000 regardless of n . The base FPR grid is fixed and log-spaced, not Monte Carlo sampled. With `n_knots=10,000`, it places 999 points in every FPR decade from 10^{-10} to 1. The fit augments that base grid with Spline 1’s high-FPR endpoint and two lowest-FPR knots, preserving the fitted boundaries and the points that define low-FPR extrapolation. These edge locations depend on the fitted sample, whereas the base allocation is fixed. Log spacing gives every order of magnitude comparable resolution and densely samples the low-FPR tail. Fixed base placement avoids random gaps in that allocation.

With `n_knots=10,000`, the production pipeline serialized under `joblib.dump` to 54–161 KB across calibration sets from 1K to 10M benign samples, remaining below 200 KB. Dependencies are `MinMaxScaler`, `IsotonicRegression`, and `Pipeline`, with no custom inference code: `joblib.load(path)` then `pipeline.predict(scores)`.

5 Evaluation on Credit Card Fraud Detection

We validate on the Credit Card Fraud Detection benchmark, which contains 284,807 European credit-card transactions and 492 fraud cases (0.172% positive rate) [8]. We load the dataset with `sklearn.datasets.fetch_openml(name="creditcard", version=1)`, using OpenML data_id 1597. A logistic-regression detector with standardized features is trained on a stratified 30% split (85,442 rows; 148 positives). The remaining 70% is the detector holdout (199,365 rows; 344 positives). A 30% stratified slice of the holdout (59,809 rows: 59,706 benign and 103 positives) serves as the calibration-fit subset. The complementary held-out-from-fit subset (139,556 rows: 139,315 benign and 241 positives) supplies the independent values in Table 3 and the primary blue curves in Figure 2. The first two figure panels show the calibration-fit subset separately for comparison. Logistic regression is chosen over a boosted-tree baseline because its linear margin resolves deep-tail FPR cleanly. The sigmoid output of a boosted-tree classifier saturates around 10^{-3} FPR on this data and truncates the ROC tail; the calibration contract holds under either detector, but only the linear detector exercises the full anchor range.

Table 3: Training and held-out FPR at calibrated score anchors. Training FPR uses the 59,706 calibration-fit benign rows; held-out FPR uses the 139,315 benign rows not used to fit calibration. Negative relative error means held-out FPR was above target.

Calibrated score	Target FPR	Raw score	Training FPR	Held-out FPR	Rel. error
0.10	10%	0.0008	9.999%	10.053%	−0.52%
0.30	1%	0.0036	0.9999%	1.023%	−2.24%
0.50	0.1%	0.0207	0.1005%	0.1005%	−0.49%
0.70	0.01%	0.9993	0.0100%	0.0093%	+7.17%

On the held-out-from-fit subset, the observed relative FPR error was at most 2.3% from 10% down to 0.1% FPR and 7.2% at 0.01% FPR. At the 0.01% target, 13 of 139,315 held-out benign rows scored at or above the corresponding raw-score threshold. This count gives an observed FPR of 0.0093% and an exact 95% Clopper–Pearson interval of 0.0050%–0.0160% [9]. The reported 7.2% describes the point-estimate error for this split, whereas the confidence interval describes its sampling uncertainty. Table 3 reports both the calibration-fit FPR and the held-out-from-fit FPR at each anchor from the latest run, and Figure 2 shows the corresponding held-out-from-fit behavior across the full curve. Because its benign rows were never touched by the calibration fit, the held-out-from-fit subset measures generalization without the fit subset’s near-target FPR-by-construction. Below 0.01% the 59,706-benign fit subset hits its binomial precision floor (Table 1), and the anchor at 10^{-5} FPR would require the calibration-fit subset to scale to about 1.5M benign samples for $\pm 50\%$ planning precision.

TPR at each fixed FPR measures detection quality, not calibration quality. The detector catches 83% of fraud at 0.1% FPR, so the calibrated operating points land on real detections rather than an empty curve. A detector whose TPR collapsed at low FPR would still calibrate to the same score axis and still fail the shipping bar. That separation is deliberate: calibration keeps the FPR at each threshold consistent across deployments while detection quality is decided separately by comparing TPR at a fixed FPR across candidate models.

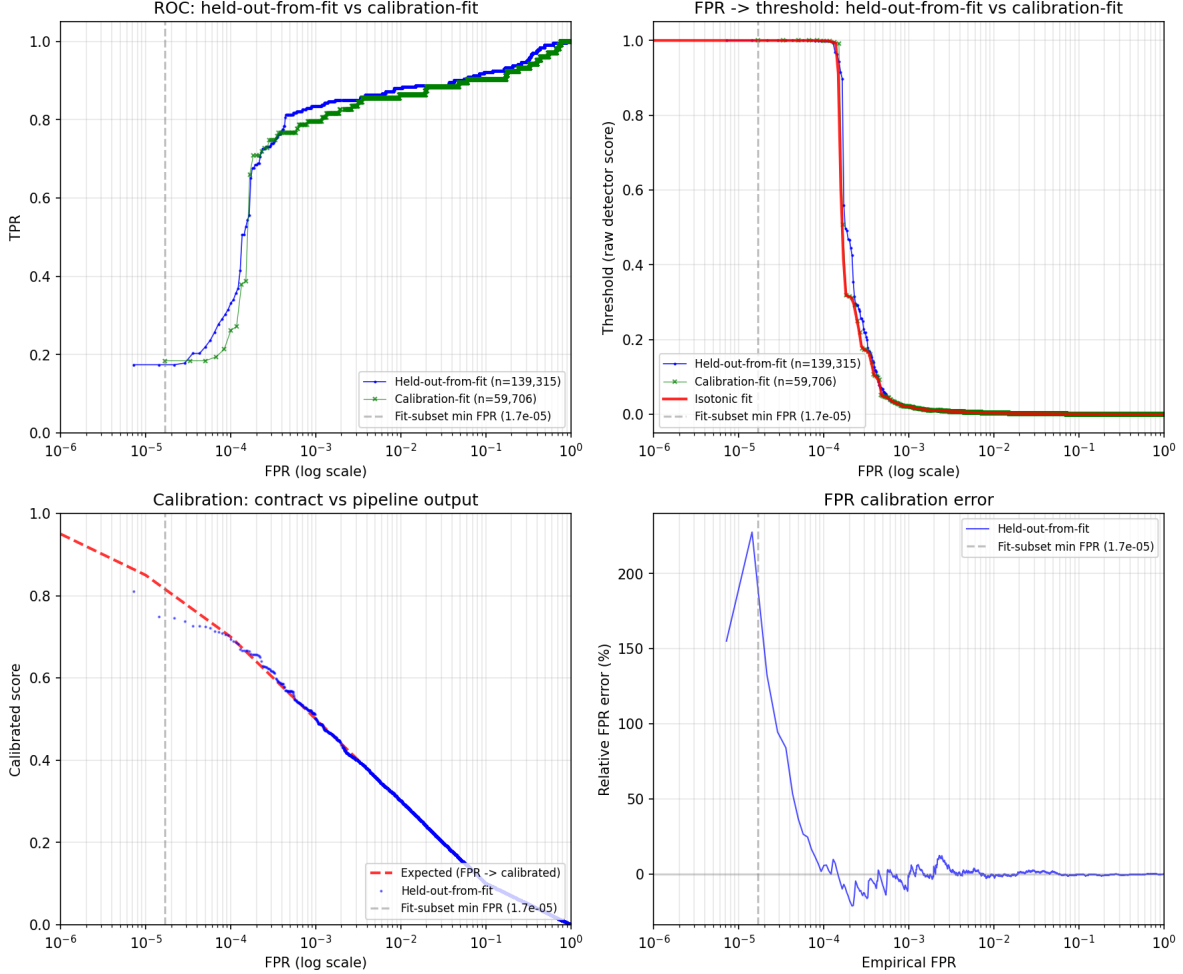


Figure 2: Four-panel diagnostics for the calibration pipeline on the Credit Card Fraud holdout. Top-left: empirical ROC (log-FPR axis) on the held-out-from-fit subset in blue and on the calibration-fit subset in green (marked with $\times$), with the fit-subset’s minimum-resolvable FPR drawn as a vertical dashed line across every panel. Top-right: the empirical FPR-to-threshold relation on the same two populations, with the red line showing the fitted FPR-to-raw-score-threshold spline. Bottom-left: expected log-scale contract (red dashed) versus pipeline output on the held-out-from-fit subset (blue). Bottom-right: relative FPR error on the held-out-from-fit subset, computed as predicted FPR minus empirical FPR divided by empirical FPR.

6 Related work

This work composes established calibration and thresholding tools into a pipeline whose released score carries an FPR meaning across the whole deployment range. Prior work supplies pieces of that construction, but not a compact score-to-FPR artifact with fixed log-FPR anchors.

Probability calibration. Platt scaling fits a sigmoid to SVM scores so they can be read as posterior class probabilities [2]. Zadrozny and Elkan use isotonic regression as a nonparametric alternative to sigmoid fitting, sorting labeled examples by score and using pool-adjacent-violators to learn a monotone step function from score to class probability [3]. Our shipped spline has the same inference shape, raw score in and calibrated value out, but the fitted quantity is the benign tail probability $q(\tau)$ from Equation (4) rather than $\Pr(Y = 1 \mid \text{score})$. This target change matters because precision moves with the positive-class rate, while FPR is defined by benign scores only.

Outlier-score normalization. Merlion includes anomaly score calibration as a post-processing module to improve interpretability in a time-series anomaly-detection library [10]. Kriegel et al. translate arbitrary

outlier factors to $[0, 1]$ values that can be compared across detectors and used in ensembles [11]. These methods output calibrated or normalized anomaly scores. Our calibrator fits the benign tail distribution directly and pins the released score to $\log_{10}(\text{FPR})$ values in Table 2.

FPR and risk control. Neyman–Pearson classifiers choose a decision rule for a specified type-I error budget. Scott and Nowak give one-sided learning-theoretic bounds of the form $R_0(\hat{h}) \leq \alpha + \epsilon$ with high probability, while Tong et al. choose an order-statistic threshold so the population type-I error exceeds α with probability at most δ [12, 13]. Learn Then Test calibrates predictive algorithms to satisfy finite-sample risk guarantees without model retraining and includes type-I outlier-error control among its examples [14]. Finn and Johnson’s early radar detector controls the threshold as a function of sampled clutter estimates, and radar textbooks treat constant-false-alarm-rate methods as threshold-detection tools [15, 16]. Inductive conformal anomaly detection computes nonconformity p-values from a proper training set and calibration scores, then raises an alarm below a chosen significance level with a false-alarm guarantee that depends on the update mode and IID assumptions [17]. Bates et al. construct conformal p-values for outlier testing and give a uniform confidence bound for FPR as the raw-score threshold varies [18]. These methods certify risk or test validity, but they do not ship a compact score transform with fixed log-FPR anchors. We estimate FPR across the score range so one released score can support block, alert, review, and logging thresholds in the same policy.

Empirical-CDF post-processing. Dadalto Câmara Gomes et al. [19] also use empirical CDFs of in-distribution detector scores, but their endpoint is a hypothesis test in which each detector score becomes a p -value under the in-distribution null and multiple detector p -values are combined with Fisher’s statistic plus Brown’s correction for correlated tests. CADET calibrates reconstruction-error anomaly scores against sample hardness and then applies a binary decision rule at an approximately tuned false-positive level [20]. Both papers calibrate anomaly or OOD scores, but their endpoint is a detector or test statistic rather than a fixed score contract for binary classifier releases.

Plotting-position corrections. Plotting positions address the rank labels used inside our first spline. The empirical label k/n is the observed calibration-set count, but a continuous benign distribution places the k -th largest score at mean fresh-sample FPR $k/(n+1)$ [5]. Filliben’s plotting positions approximate the median rank location and, after flipping from lower-tail rank to upper-tail FPR, give median-centered FPR labels [6, 7]. These corrections are upstream knot-label choices for Section 2.2, not competing calibrators.

Pipeline composition. The isotonic primitive, empirical-CDF transform, plotting-position correction, and monotone anchoring each predate this work. The pipeline combines whole-curve FPR calibration on benign scores, a fixed piecewise-linear output contract in $\log_{10}(\text{FPR})$ space, a deterministic artifact under 200 KB in our measurements, below-floor extrapolation, and release packaging that ships the calibrator alongside the model.

7 Limitations

Score-granularity floor. A threshold cannot separate samples with the same score. Here, 24 benign samples share the maximum score, so the detector jumps from zero FPR to $24/59,706 = 0.0402\%$ FPR.

Extrapolation below the sample-size floor. Calibration below the fit set’s lowest observed tail FPR is linear extrapolation from the two lowest observed points. For production FPR targets below the floor, fit calibration on a larger benign set rather than rely on extrapolation.

Benign distribution drift. The stability of the FPR contract assumes production benign traffic resembles calibration-time benign traffic. Drift in legitimate user behavior shifts every threshold’s actual FPR. This is a data-freshness concern, not a defect of the calibrator, and it applies regardless of calibration method.

Plotting-position refinement. The implementation uses Filliben’s median plotting position [6] by default and exposes the mean-centered $k/(n+1)$ position as an option. This adjustment fixes the FPR label assigned to rank-selected thresholds, but it does not reduce the finite-sample spread, so it does not change the sample-size requirements in Table 1.

8 Usage

Fitting, offline:

```
from fpr_model_calibration.calibration import fit_calibration_pipeline
```

```
import joblib

pipeline = fit_calibration_pipeline(benign_scores, n_knots=10000)
joblib.dump(pipeline, 'calibration.pkl')
```

Inference, production:

```
import joblib

pipeline = joblib.load('calibration.pkl')
calibrated = pipeline.predict(scores.reshape(-1, 1))
```

The log anchors ($0.5 = 0.1\%$ FPR, $0.7 = 0.01\%$, and so on) are tuned to AI security, where 1-in-1,000 is a practical planning point for automated action. For domains with a different operational block threshold, edit `_FPR_CAL_KNOTS` in `src/fpr_model_calibration/calibration.py` so that 0.5 sits at the target FPR.

9 Conclusion

FPR calibration should give operators a stable alert-rate meaning for every score threshold, not only a calibrated probability at one operating point. This method does that with benign-only data, a fixed log-scale FPR-to-calibrated-score contract, and two sklearn linear splines packaged as a standard `Pipeline`. The first spline converts target FPRs to rescaled-score thresholds during fitting. The second spline is the shipped raw-score-to-calibrated-score pipeline used at inference.

The evaluation shows that a fixed 10K-knot artifact preserves the calibrated curve when the benign calibration set supports the target FPR range, with the serialized artifact staying below 200 KB in our measurements.

The main constraint is benign sample count. Filliben plotting positions place first-spline knots more accurately, but the dominant low-FPR error is still finite-sample uncertainty. The 10K-knot representation makes deployment cheap; for production targets below the observed FPR floor, the practical answer is more benign calibration data, not a more elaborate spline.

Acknowledgments

This work was prepared with AI assistance. AI coding assistants and large language models were used for code review and refactoring of the accompanying software, prose editing of the manuscript, and citation verification. The author reviewed and verified all technical content, results, and final wording, and bears full responsibility for the work.

References

- [1] Adam Swanda, Amy Chang, Alexander Chen, Fraser Burch, Paul Kassianik, and Konstantin Berlin. A framework for rapidly developing and deploying protection against large language model attacks. In *Proceedings of the 2025 Conference on Applied Machine Learning for Information Security (CAMLIS)*, volume 299 of *Proceedings of Machine Learning Research*, pages 200–221, Arlington, VA, USA, 2025. PMLR. doi: 10.48550/arXiv.2509.20639. URL <https://proceedings.mlr.press/v299/swanda25a.html>.
- [2] John C. Platt. Probabilities for SV machines. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large-Margin Classifiers*, pages 61–74. MIT Press, Cambridge, MA, 2000. ISBN 9780262194488. doi: 10.7551/mitpress/1113.003.0008.
- [3] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 694–699, Edmonton, Alberta, Canada, 2002. ACM. ISBN 158113567X. doi: 10.1145/775047.775151.
- [4] NIST/SEMATECH. Confidence intervals. e-Handbook of Statistical Methods, Section 7.2.4.1, n.d.. URL <https://www.itl.nist.gov/div898/handbook/prc/section2/prc241.htm>. Accessed 2026-05-28.
- [5] NIST/SEMATECH. Order statistics means. Dataplot Reference Manual, n.d.. URL <https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/ordstmea.htm>. Accessed 2026-05-07.

- [6] James J. Filliben. The probability plot correlation coefficient test for normality. *Technometrics*, 17(1): 111–117, 1975. ISSN 0040-1706. doi: 10.1080/00401706.1975.10489279.
- [7] NIST/SEMATECH. Probability plot. e-Handbook of Statistical Methods, n.d.. URL <https://www.itl.nist.gov/div898/handbook/eda/section3/probplot.htm>. Accessed 2026-05-07.
- [8] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 159–166, Cape Town, South Africa, 2015. IEEE. ISBN 978-1-4799-7560-0. doi: 10.1109/SSCI.2015.33.
- [9] C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, December 1934. ISSN 0006-3444. doi: 10.1093/biomet/26.4.404.
- [10] Aadyot Bhatnagar, Paul Kassianik, Chenghao Liu, Tian Lan, Wenzhuo Yang, Rowan Cassius, Doyen Sahoo, Devansh Arpit, Sri Subramanian, Gerald Woo, Amrita Saha, Arun Kumar Jagota, Gokulakrishnan Gopalakrishnan, Manpreet Singh, K. C. Krithika, Sukumar Maddineni, Daeki Cho, Bo Zong, Yingbo Zhou, Caiming Xiong, Silvio Savarese, Steven Hoi, and Huan Wang. Merlion: End-to-end machine learning for time series. *Journal of Machine Learning Research*, 24(226):1–6, 2023. ISSN 1533-7928. URL <http://jmlr.org/papers/v24/22-0809.html>.
- [11] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Interpreting and unifying outlier scores. In *Proceedings of the 11th SIAM International Conference on Data Mining (SDM)*, pages 13–24, Mesa, AZ, USA, 2011. SIAM. ISBN 9780898719925. doi: 10.1137/1.9781611972818.2.
- [12] Clayton Scott and Robert Nowak. A Neyman–Pearson approach to statistical learning. *IEEE Transactions on Information Theory*, 51(11):3806–3819, 2005. ISSN 0018-9448. doi: 10.1109/TIT.2005.856955.
- [13] Xin Tong, Yang Feng, and Jingyi Jessica Li. Neyman–Pearson classification algorithms and NP receiver operating characteristics. *Science Advances*, 4(2):eaao1659, 2018. ISSN 2375-2548. doi: 10.1126/sciadv.aao1659.
- [14] Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, June 2025. ISSN 1932-6157. doi: 10.1214/24-AOAS1998.
- [15] H. M. Finn and R. S. Johnson. Adaptive detection mode with threshold control as a function of spatially sampled clutter-level estimates. *RCA Review*, 29(3):414–464, September 1968.
- [16] Mark A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill Education, New York, NY, 2nd edition, 2014. ISBN 9780071798327.
- [17] Rikard Laxhammar and Göran Falkman. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, 74(1–2):67–94, 2015. ISSN 1012-2443. doi: 10.1007/s10472-013-9381-7.
- [18] Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p -values. *The Annals of Statistics*, 51(1):149–178, February 2023. ISSN 0090-5364. doi: 10.1214/22-AOS2244.
- [19] Eduardo Dadalto Câmara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Combine and conquer: A meta-analysis on data shift and out-of-distribution detection. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. doi: 10.48550/arXiv.2406.16045. URL <https://openreview.net/forum?id=VGNBUS9TrU>.
- [20] Ailin Deng, Adam Goodge, Lang Yi Ang, and Bryan Hooi. CADET: Calibrated anomaly detection for mitigating hardness bias. In Luc De Raedt, editor, *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2002–2008, Vienna, Austria, 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi: 10.24963/ijcai.2022/278.

A Sample-size planning rule

The sample-size table uses the normal approximation in Equation (2) as a planning rule. Use the Clopper–Pearson interval [9] for exact intervals. At target FPR p , the 95% half-width is approximately

$$2\sqrt{\frac{p(1-p)}{n}}. \quad (\text{A.1})$$

In the low-FPR regime, $p(1 - p) \approx p$, so the half-width becomes

$$2\sqrt{\frac{p}{n}}. \quad (\text{A.2})$$

A relative half-width of r means that the absolute half-width is at most rp . Setting Equation (A.2) equal to rp and solving gives

$$2\sqrt{\frac{p}{n}} = rp \implies n = \frac{4}{r^2 p}. \quad (\text{A.3})$$

For example, $p = 10^{-3}$ and $r = 0.5$ gives $n = 16,000$ benign samples.

B Plotting-position simulation

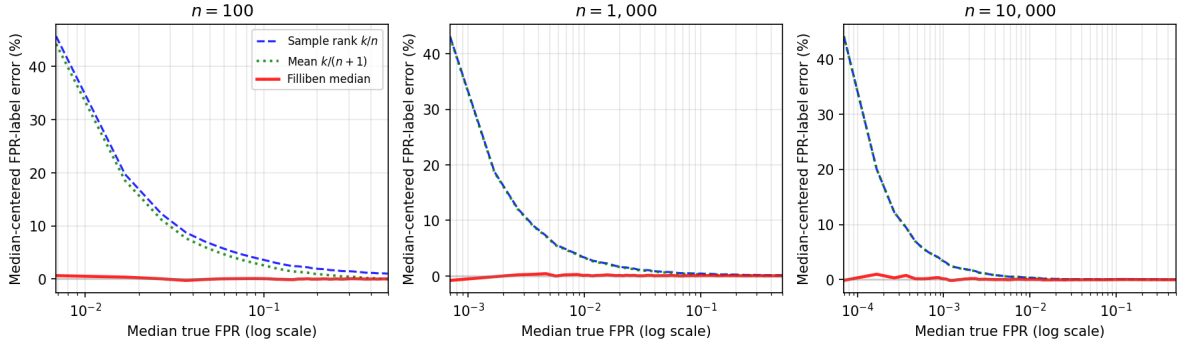


Figure B.1: Median-centered FPR-label error for threshold τ_k , the k -th largest benign model score. Each experiment draws n benign scores from $\text{Uniform}(0, 1)$, selects the threshold τ_k , and computes its fresh-sample FPR as $q_k = 1 - \tau_k$. Because the uniform distribution has survival function $\Pr(S \geq \tau) = 1 - \tau$, this measures q_k directly rather than estimating it from another finite sample. For each plotting-position rule $m \in \{\text{sample}, \text{mean}, \text{Filliben}\}$, the plotted error is $\tilde{q}_k^m - \text{median}(q_k)$, divided by $\text{median}(q_k)$, across 50,000 repeated calibration samples.

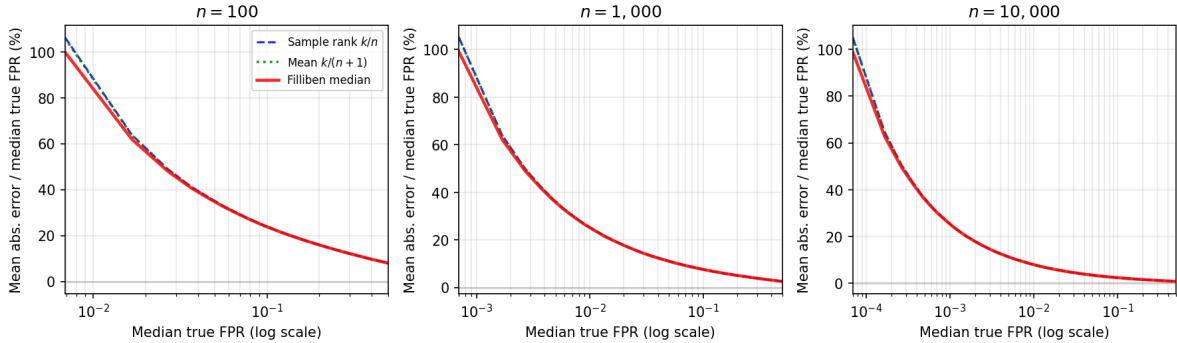


Figure B.2: Mean absolute relative FPR-label error for the same simulation as Figure B.1. For each plotting-position rule m and rank k , the plotted value is the average of $|\tilde{q}_k^m - q_k|$ across repeated calibration samples, divided by the median fresh-sample FPR for that rank. The denominator uses the median fresh-sample FPR rather than the per-draw fresh-sample FPR because rare maximum-score draws can make q_k arbitrarily close to zero. This plot includes both the label-centering error and the finite-sample spread of the rank-selected threshold.