Evaluating Performance, Safety, and Robustness of an AI-Based Airport Delay Alerting Tool with Calibrated Machine Learning for Operational Decision Support

Soufiane Momtaz
ENSET Mohammedia, Hassan II University
Casablanca, Morocco
Email: soufiane.momtaz@gmail.com

Otmane Idrissi
ENSET Mohammedia, Hassan II University
Casablanca, Morocco
Email: iodrimane@gmail.com

Joseph Machrouh
Thales LAS France
Rungis, France
Email: joseph.machrouh@thalesgroup.com

***Abstract*—Artificial intelligence (AI)-based decision-support tools in airport operations should be assessed not only by predictive performance but also by the safety and robustness of the alerting policies they induce under temporal drift. This paper presents a leakage-free one-month-ahead case study on United States (U.S.) Bureau of Transportation Statistics (BTS) airport-carrier-month data from 2022–2024 and makes four contributions. First, this work specifies a system pathway from pre-month data to calibrated probabilistic scoring, thresholded alerting, and deployment audit. Second, this work introduces an assurance-oriented evaluation protocol that combines the area under the receiver operating characteristic curve (AUC), Brier score, and log-loss for prediction; nominal-threshold decision risk for safety; and Threshold-Local Calibration Error (TLCE), Action-Overconfidence Gap (AOG), and risk stability across thresholds for robustness. Third, this work compares logistic regression, random forest, and extreme gradient boosting (XGBoost) under a strict train–calibration–test chronology and evaluates Platt, isotonic, and beta calibration under temporal transfer, positioning these tabular baselines against state-of-the-art delay-prediction practice. Fourth, this work discusses scalability and security-aware operational integration and shows through monthly audit that calibration behavior changes across deployment sub-regimes. The main conclusion is that calibration is a time-sensitive operational component that must be validated through local diagnostics, threshold-aware reporting, and continuous deployment auditing.**

Keywords-safety; robustness; performance; artificial-intelligence-based systems; temporal drift; calibration; airport delay alerting.

I. Introduction

Artificial intelligence (AI)-based systems are increasingly used to issue alerts, prioritize interventions, and support planning in airport and transportation operations. Once a probabilistic score is thresholded into an action, evaluation should cover not only ranking ability but also the operational consequences of acting or not acting on that score. In this paper, *performance* denotes predictive quality, *safety* denotes cost-weighted decision behavior under missed and unnecessary alerts, and *robustness* denotes stability across thresholds and changing operating conditions. The aim is to support assurance-oriented evaluation of AI-based decision-support tools for aviation operations, not to claim certification of the studied prototype.

Operational evaluation should separate probabilistic scores from the action rules and costs they induce. Threshold-insensitive summaries, such as the area under the receiver operating characteristic curve (AUC), can hide materially different operational behaviors [1]. Calibration and probabilistic-forecasting studies also show that strong discrimination does not guarantee reliable probabilities [2]–[9]. Under dataset shift, uncertainty quality can deteriorate, which is a major issue for the safe operational use of AI systems [10].

This paper revisits airport delay analysis from an operational decision-making standpoint using United States (U.S.) Bureau of Transportation Statistics (BTS) data. The task is formulated as *one-month-ahead* high-delay alerting using lagged information, and the main leakage path is removed by excluding target-month operational outcomes from the feature set. The evaluation protocol is chronological: models are trained on 2022–2023, calibration maps are fitted on the first half of 2024 (2024-H1), and forward performance is measured on the second half of 2024 (2024-H2). Four main contributions are made. First, this work defines the alerting system pathway and frames airport delay regime forecasting as a performance-safety-robustness problem. Second, this work uses assurance indicators targeted at the action threshold rather than global accuracy alone. Third, this work compares three model families and three post-hoc calibration strategies under explicit temporal transfer and positions the evaluation against state-of-the-art delay-prediction practice. Fourth, this work adds scalability considerations and a monthly deployment audit showing that the calibration effect can reverse across sub-regimes within the same deployment window.

This study treats calibration as an operational control variable whose benefit depends on model family, policy threshold, cost ratio, and time elapsed between calibration and deployment.

The rest of the paper is structured as follows. In Section 2, we review decision-centric evaluation, calibration, and airport-delay prediction. In Section 3, we define the assurance framework. In Section 4, we describe the task, threat model, data, and chronology. In Section 5, we present features, models, calibration maps, and the bootstrap protocol. In Section 6, we report results. In Section 7, we discuss assurance implications, scalability, security, and limitations. In Section 8, we present deployment scenarios and monitoring checks. Finally, in Section 9, we conclude the work.

II. Related Work and Positioning

Previous studies have criticized classifier evaluation for mixing ranking quality with decision quality and show that summaries, such as AUC, can hide the threshold and cost structure governing operational use [1]. This is crucial for

alerting tools, where a model score becomes an action through a governed threshold rather than a purely statistical comparison.

Calibration-related research asks whether predicted probabilities are consistent with observed frequencies. In machine learning, Platt scaling, isotonic regression, and beta calibration show that good discrimination does not imply good probabilities [4][5][7]. Calibration conclusions also depend on the diagnostic being used [9], which motivates threshold-local calibration rather than reliance on a single global score.

The present operational setting differs from abstention-based evaluation: the airport alerting tool should either trigger an alert or remain silent at a governed policy threshold. For that reason, local calibration and cost-weighted decision risk, rather than coverage guarantees alone, are the primary assurance objects here.

Robust deployment adds another element: benchmark studies show that uncertainty quality and calibration can degrade materially when the deployment distribution changes [10]. This motivates ongoing monitoring and validation rather than one-off test-set reporting, and this paper follows that logic by treating the calibrator as a time-sensitive component inside the assurance loop.

In the aviation literature, and particularly in airport operations, most delay studies have been framed as prediction problems rather than assurance problems. A recent review shows that the field has focused primarily on improving forecast accuracy across different scopes, data sources, and horizons [11]. Aviation studies also examine recurrent learners such as long short-term memory (LSTM) and convolutional neural network–LSTM hybrids, which are natural candidates when flight-level trajectories, daily sequences, or airport-network time series are available [12][13]. Relative to this state of the art, the present comparison is not a leaderboard over heterogeneous datasets and targets; it evaluates representative tabular learners under a common leakage-free BTS task and asks whether their thresholded alerting policies remain safe and robust under calibration transfer and temporal drift.

III. Decision-Centric System and Assurance Framework

The operational system is a monthly alerting pipeline: a data snapshot available up to month $t-1$ is transformed into lagged and rolling features, a model estimates the probability $\hat{p}_t$ of a high-delay regime in month t, an optional calibrator fitted on a later validation window adjusts the score, a governed threshold converts the score into an alert, and a deployment audit checks prevalence, action rate, local calibration, and decision risk. This system view clarifies that the object being assessed is not only a classifier, but the full score-to-action pathway used by a human-supervised decision-support tool.

The assurance layer is organized around the three dimensions presented in Table I. Predictive performance evaluates whether the model can distinguish upcoming high-delay regimes. Safety evaluates the cost incurred by the alerting policy at the governed operating point. Robustness evaluates whether the performance and safety conclusions remain stable when the threshold, the cost ratio, or the temporal regime changes. A model can perform well on one dimension and poorly on another, which motivates the separation of these layers rather than treating "accuracy" as a sufficient surrogate for deployment quality.

TABLE I. Assurance dimensions used in the study.

Dimension	Main indicators	Operational question
Performance	AUC, Brier, log-loss	Can the model predict next-month high-delay regimes?
Safety	$R(\tau^*)$, action rate	What is the cost of the current policy threshold?
Robustness	local calibration, action gap, mean $R(T)$, S_R	Does behavior remain trustworthy when the regime or threshold moves?

Let $\hat{p}(x) \in [0,1]$ denote the predicted probability of a high-delay regime and let

$$d_\tau(x) = \mathbb{1}\{\hat{p}(x) \geq \tau\} \tag{1}$$

be the alerting policy. With false-positive cost $C(1,0) = 1$, false-negative cost $C(0,1) = 5$, and zero cost for correct decisions, the nominal Bayes threshold under calibrated probabilities is

$$\tau^* = \frac{1}{1+5} = \frac{1}{6} \approx 0.167. \tag{2}$$

The expected decision risk is

$$R(\tau) = \mathbb{E}[C(d_\tau(X), Y)], \tag{3}$$

reported per unweighted airport-carrier-month decision. This evaluates policy consistency across decision units; flight- or passenger-weighted risk would answer a different operational-impact question and is treated as future work. The 5:1 asymmetry is a nominal cost scenario rather than a validated safety-cost model, so Section VI-C also stress-tests 2:1 and 10:1 ratios. Threshold-local calibration error (TLCE) is measured by

$$\mathrm{TLCE}_h(\tau) = \left|\mathbb{E}[Y - \hat{p}(X) \mid |\hat{p}(X) - \tau| \leq h]\right|, \tag{4}$$

where $h = 0.05$ is a pre-specified five-percentage-point half-width around the governed threshold, creating a 0.10-wide local audit band. It focuses the audit on near-threshold decisions while keeping enough observations for a stable local estimate; calibration diagnostics depend on such neighborhood choices [9]. The action-overconfidence gap (AOG) is

$$\mathrm{AOG}(\tau) = \mathbb{E}[\hat{p}(X) - Y \mid \hat{p}(X) \geq \tau]. \tag{5}$$

A positive AOG indicates optimism on the cases that actually trigger alerts. Threshold robustness is evaluated on the grid $T = \{0.05, 0.06, \ldots, 0.60\}$ using mean threshold-averaged risk and

$$S_R(T) = \mathrm{Std}_{\tau \in T} R(\tau). \tag{6}$$

This framework explains why local calibration matters when decisions are taken locally and not globally.

IV. Operational Setting, Threat Model, and Data

A. Operational task and threat model

In this study, we use the U.S. Bureau of Transportation Statistics (BTS) Airline Delay Cause dataset [14], which contains 68,194 airport-carrier-month observations from January 2022 to December 2024, covering 377 U.S. airports, 21 airlines, and 36 calendar months. After removing 109 rows with missing target fields, 68,085 observations remain in the study.

For each airport-carrier-month tuple, the binary event is

$$Y_t = \mathbb{1}\left\{ \frac{\texttt{arr\_del15}_t}{\texttt{arr\_flights}_t} \geq 0.25 \right\}, \tag{7}$$

which marks a monthly high-delay regime. Operationally, $d_\tau(x) = 1$ means issuing a high-delay alert for the coming month to trigger heightened monitoring or mitigation planning. BTS defines an arrival delay indicator using the standard 15-minute-or-more lateness rule [14]; the 25% monthly cutoff is therefore a pre-specified study threshold corresponding to at least one quarter of arrivals being delayed for an airport-carrier month. It is not claimed as a universal safety threshold, but as an empirical marker of sustained monthly degradation rather than isolated daily disruption, and it should therefore be redefined and revalidated before transfer to other regions, traffic mixes, or governance contexts.

Three failure modes are considered, each tied to a known assurance risk. The first is "contemporaneous leakage": if same-month delays, cancellations, or delay minutes are included, the model becomes a partly contemporaneous classifier rather than a forward operational alerting tool, a leakage pattern known to overstate deployment performance and to weaken data-integrity claims at deployment [15]. The second is "calibration-transfer failure": a calibration map fitted on one window may become unreliable when the regime changes, as expected under uncertainty degradation during distribution shift [10]. The third is "threshold fragility": a model can behave well at one threshold but become unstable or costly when the operating point changes, reflecting the dependence of classifier utility on costs and thresholds [1].

B. Chronological protocol and regime variation

The temporal split is summarized in Table II. Training uses 45,498 observations from 2022–2023. Calibration uses 11,291 observations from the first half of 2024 (2024-H1). The forward test uses 11,296 observations from the second half of 2024 (2024-H2). Figure 1 shows why this split is demanding: the monthly prevalence of the high-delay regime varies materially inside 2024. A calibration map fitted in the first half of 2024 therefore faces a transfer problem in the second half of the same year.

TABLE II. Chronological split summary.

Split	Period	n	Positive rate
Train	2022-01 to 2023-12	45,498	0.290
Calibration	2024-01 to 2024-06	11,291	0.304
Test	2024-07 to 2024-12	11,296	0.266

This split governs all experiments.

V. Feature Engineering and Experimental Design

A. Leakage-free feature construction

Only information available before month t is used to forecast Y_t. A full monthly panel is constructed for all observed airport-carrier pairs, and lagged and rolling statistics are computed along each pair's time axis. The last supervised rows are only those months that are actually observed in the BTS table, so artificial panel completion is not used to create target-month information.

Table III summarizes the feature families. The model sees historical regime state, disruption composition, exposure, volatility, and persistent entity effects but it does not see any same-month outcomes from the target month. Exogenous forecasts, such as weather or network-state outlooks, are excluded in this first assurance case to preserve a reproducible leakage-free BTS-only protocol. The resulting problem is a genuine forward alerting task, not a disguised same-month classifier.

TABLE III. Leakage-free feature families used for one-month-ahead forecasting.

Feature	Examples	Pre-t?	Role
Historical regime state	Lagged delay rate; delay minutes per flight at lags 1, 2, 3, and 6	Yes	Captures persistence and recovery after prior disruptions
Disruption composition	Lagged carrier, weather, National Airspace System (NAS), security, and late-aircraft counts and minutes per flight	Yes	Separates heterogeneous precursor mechanisms behind future delay regimes
Exposure and scale	Lagged log flight volume; cancellation and diversion rates	Yes	Represents traffic pressure and exposure to disruption accumulation
Volatility and memory	Rolling means and standard deviations over earlier months	Yes	Captures whether the airport-carrier pair is entering an unstable operating regime
Seasonality and entity effects	Month sine/cosine terms; airport and carrier encodings	Yes	Represents recurring seasonal structure and persistent airport/carrier heterogeneity

B. Models, calibration maps, and bootstrap protocol

Three baseline model families are compared: logistic regression, random forest (RF), and extreme gradient boosting, implemented as XGBoost (XGB) [16][17]. They represent an interpretable linear model, a bagged tree ensemble, and a scalable gradient-boosted tree baseline widely used for tabular prediction. Recurrent networks were considered conceptually but not benchmarked here because the decision unit is a short monthly airport-carrier panel; a fair LSTM study would require longer flight-level or daily sequences, leakage-safe sequence

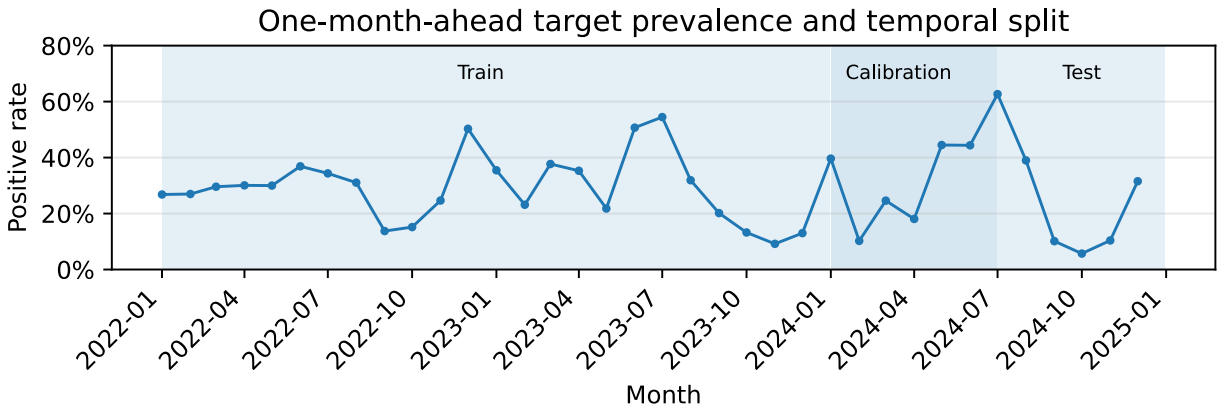

Figure 1. Monthly prevalence of the high-delay regime with chronological train, calibration, and test segments. The 2024 calibration and test windows exhibit materially different operating conditions.

pooling, and the same calibration-transfer audit. Hyperparameters were kept moderate and fixed to emphasize evaluation rather than leaderboard optimization: logistic regression with L2 regularization, random forest with 40 trees (maximum depth 14, minimum leaf size 5, 80% row subsampling), and XGBoost with 250 trees of depth 5, learning rate 0.05, and 80% row and column subsampling.

For the two tree-based models, three post-hoc calibration maps are fitted on the first half of 2024 (H1) calibration window only: Platt scaling [4], isotonic regression [5], and beta calibration [7]. Predictive performance is measured by AUC, Brier score [3], and log-loss. For selected pairwise comparisons, 400 bootstrap resamples of the second half of 2024 (H2) test set provide confidence intervals.

VI. Results

A. Forward predictive performance and nominal-threshold safety

For compact tables and figures, RF denotes random forest and XGB denotes XGBoost. Table IV reports the uncalibrated baselines on the forward test. XGBoost is strongest on all predictive metrics and on the nominal operating point. It achieves AUC 0.811, Brier 0.145, log-loss 0.450, and $R(\tau^*) = 0.515$. Random forest is second best, with AUC 0.768 and $R(\tau^*) = 0.572$, while logistic regression trails slightly on both fronts. Bootstrap intervals for the strongest model are tight: XGBoost reaches AUC 0.8113 with 95% interval [0.8026, 0.8204] and nominal-threshold risk 0.5149 with interval [0.4933, 0.5335].

TABLE IV. Forward-test performance and nominal-threshold safety on 2024-H2.

Model	AUC	Brier	Log-loss	$R(\tau^*)$
Logistic regression	0.763	0.161	0.493	0.612
Random forest	0.768	0.157	0.483	0.572
XGBoost	0.811	0.145	0.450	0.515

The baseline ranking is operationally relevant: if one had to select a single uncalibrated family for deployment under the 5:1 cost ratio, XGBoost would be preferred. However, this is only the first layer of the analysis. The next issue is whether calibration improves or degrades the safety case once it is learned on one time window and applied later under drift.

B. Calibration transfer under drift

Table V compares the tree families before and after post-hoc calibration. The first result is straightforward: all three calibrators improve the 2024-H1 calibration window. For XGBoost, Platt scaling reduces $\mathrm{TLCE}_{0.05}(\tau^*)$ from 0.1018 to 0.0134. For random forest, the corresponding value falls from 0.0550 to 0.0020. Isotonic regression drives the calibration-window local error effectively to zero for both families, which is a reminder that highly flexible calibrators can fit the source window extremely closely.

The forward-test behavior is different. For XGBoost, all three static calibrators slightly worsen safety and robustness on 2024-H2. Platt scaling increases TLCE from 0.0348 to 0.0441, increases $R(\tau^*)$ from 0.5149 to 0.5266, and increases mean threshold-averaged risk from 0.6766 to 0.6928. For random forest, the pattern is mixed: Platt scaling increases $R(\tau^*)$ from 0.5718 to 0.6017, but it lowers mean threshold-averaged risk from 0.7552 to 0.7254 and lowers S_R from 0.1595 to 0.1390. Beta calibration closely tracks Platt in this dataset, while isotonic regression is the least stable transfer option.

Figure 2 visualizes the transfer behavior for XGBoost. On the H1 calibration window, Platt scaling aligns the model much more closely with the diagonal in the operating region below 0.4. On the H2 forward test, the same calibration map no longer dominates the raw model near τ^*. The lesson is not that calibration is useless. It is that calibration itself is time-sensitive and must be validated as part of the decision policy rather than assumed safe once fitted.

To isolate the transfer mechanism further, Table VI reports selected source-window and deployment-window diagnostics, including expected calibration error (ECE), for the raw and Platt-scaled models. For both tree families, ECE and local error improve on H1 as intended. Yet the deployment consequences differ. Random forest preserves almost the same action rate

TABLE V. SAFETY AND ROBUSTNESS TRANSFER FROM 2024-H1 CALIBRATION TO THE 2024-H2 FORWARD TEST. THRESHOLD-LOCAL CALIBRATION ERROR USES $h = 0.05$ AROUND $\tau^* = 1/6$.

Family	Calibration	TLCE H1	TLCE H2	AOG H2	Action H2	$R(\tau^*)$ H2	Mean $R(T)$ H2	$S_R(T)$ H2
Random forest	None	0.0550	0.0169	0.0034	0.6423	0.5718	0.7552	0.1595
Random forest	Platt	0.0020	0.0554	0.0506	0.7529	0.6017	0.7254	0.1390
Random forest	Beta	0.0024	0.0546	0.0508	0.7523	0.6022	0.7250	0.1392
Random forest	Isotonic	0.0000	0.0636	0.0553	0.7289	0.5995	0.7373	0.1652
XGBoost	None	0.1018	0.0348	-0.0203	0.4807	0.5149	0.6766	0.1454
XGBoost	Platt	0.0134	0.0441	-0.0018	0.6561	0.5266	0.6928	0.1672
XGBoost	Beta	0.0119	0.0444	-0.0025	0.6586	0.5271	0.6966	0.1668
XGBoost	Isotonic	0.0000	0.0598	0.0026	0.6575	0.5270	0.7036	0.1773

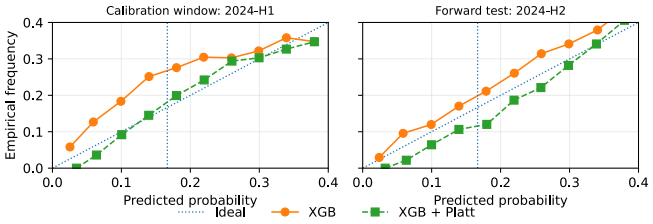

Figure 2. Threshold-local reliability for XGBoost before and after Platt scaling. Calibration improves the 2024-H1 window on which it is fitted, but the same map does not transfer cleanly to the 2024-H2 forward test.

inflation from H1 to H2, while XGBoost shifts from 0.5699 to 0.7433 on H1 and from 0.4807 to 0.6561 on H2. The transfer failure is therefore not merely about local score fit; it also changes how aggressively the policy fires.

TABLE VI. SELECTED TRANSFER DIAGNOSTICS FOR RAW AND PLATT-SCALED MODELS.

Model	ECE H1	ECE H2	Act. H1	Act. H2	R H1	R H2
RF	0.045	0.011	0.638	0.642	0.630	0.572
RF + Platt	0.015	0.050	0.753	0.753	0.609	0.602
XGB	0.064	0.030	0.570	0.481	0.661	0.515
XGB + Platt	0.021	0.038	0.743	0.656	0.596	0.527

C. Threshold robustness and cost-ratio sensitivity

Figure 3 shows forward-test risk curves across thresholds. XGBoost is the strongest family overall, and its raw probabilities already yield the best mean risk profile under the 5:1 cost ratio. By contrast, calibrated random-forest variants move the curve downward on average even while degrading the nominal-threshold point. This result shows why safety and robustness must be reported together: depending on whether deployment uses a fixed threshold or a range of plausible thresholds, the same calibration map can appear harmful or helpful.

Bootstrap paired differences make the point sharper. Relative to raw XGBoost, Platt scaling increases mean threshold-averaged risk by 0.0162 with 95% interval [0.0120, 0.0206]. Relative to raw random forest, Platt scaling reduces mean threshold-averaged risk by 0.0297 with interval [-0.0334, -0.0260] while increasing nominal-threshold risk by 0.0297 with interval [0.0171, 0.0418]. The sign of the calibration effect is therefore policy-dependent.

Figure 4 extends the analysis to false-negative to false-positive cost ratios of 2:1, 5:1, and 10:1. For XGBoost, Platt scaling is slightly better at the nominal threshold for 2:1 and 10:1, but it is worse in mean threshold-averaged risk for all

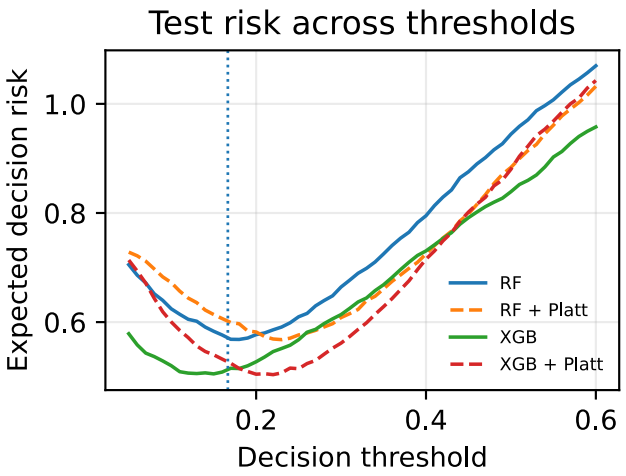

Figure 3. Forward-test risk curves across thresholds for the two tree families with and without Platt scaling. The dotted line marks the nominal Bayes threshold $\tau^* = 1/6$ for the 5:1 cost ratio.

three ratios. For random forest, Platt scaling increases nominal-threshold risk for all three ratios, yet decreases mean threshold-averaged risk for all three. The optimal threshold also shifts materially with the cost ratio, which means that model selection and calibration cannot be separated from policy design.

D. Monthly deployment audit

A deployment-oriented view is obtained by auditing the test months individually. Figure 5 shows that the effect of calibration changes with the regime for both tree families. For XGBoost in July 2024, when prevalence is 62.7%, Platt scaling helps despite more aggressive alerting: risk falls from 0.639 to 0.465 while the action rate rises from 0.742 to 0.860. In October 2024, when prevalence collapses to 5.7%, the same calibration map over-triggers alerts and harms safety: risk rises from 0.285 to 0.427 while the action rate rises from 0.259 to 0.452. Random forest shows a related but distinct pattern: calibration persistently raises the action rate, and it remains riskier in low-prevalence months, such as October and November, even when its threshold-averaged profile is smoother overall.

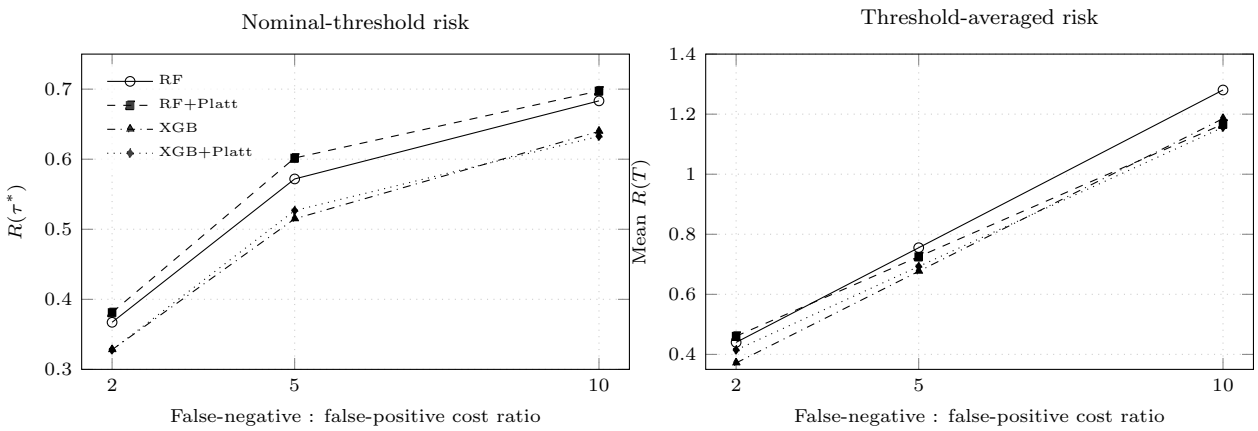

Figure 4. Sensitivity of safety conclusions to the false-negative to false-positive cost ratio. The left panel uses the nominal Bayes threshold for each ratio; the right panel averages risk over the threshold range T. Distinct markers and line types distinguish model variants for readability in print.

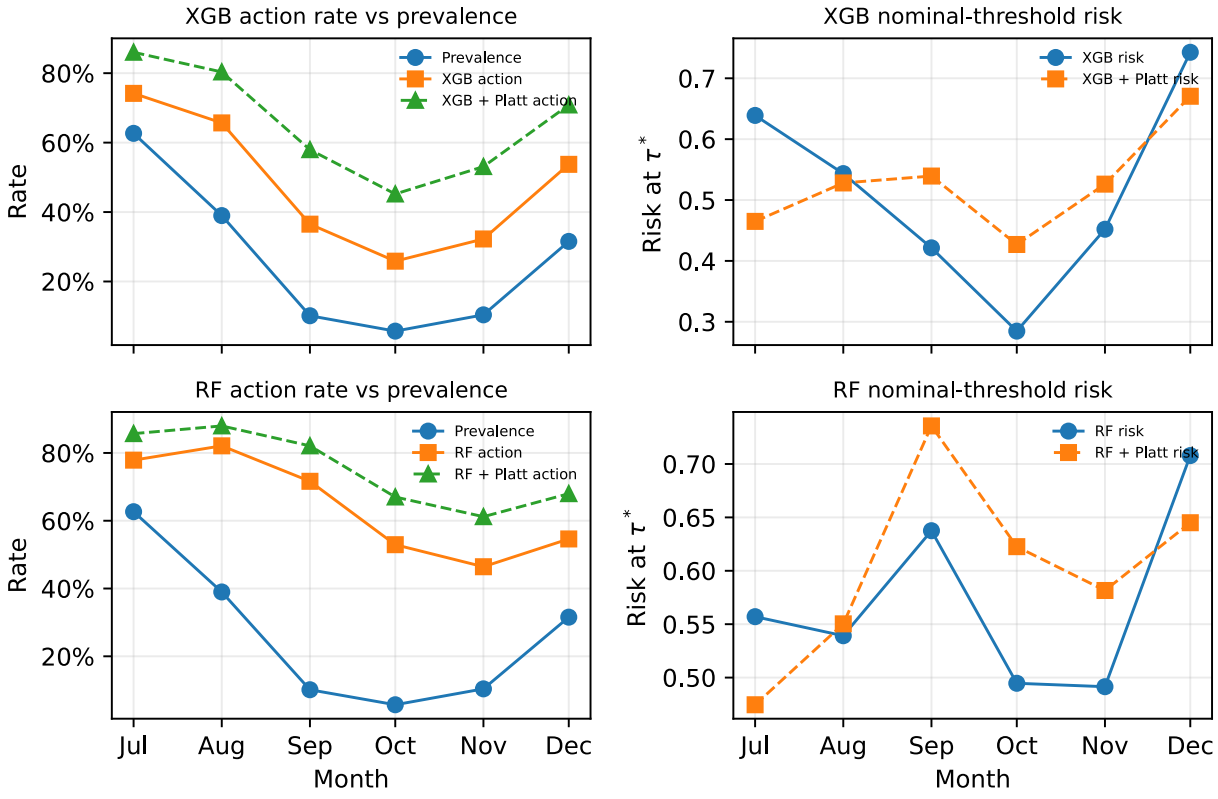

Figure 5. Monthly deployment audit on the 2024-H2 deployment window. For both tree families, the interaction between prevalence, action rate, and nominal-threshold risk changes across months, which is why monthly safety auditing is more informative than a single aggregate score.

 ISBN: 978-1-68558-395-8

VII. Discussion and Operational Assurance Implications

The leakage-free forward protocol changes the interpretation in a meaningful way. A simpler story would have been that calibration lowers decision risk. The stronger result is more nuanced and more credible: calibration is an operational element whose value depends on transfer conditions. Under drift, a calibration map can improve the source window while degrading the deployment window. For an operational aviation audience, this means calibration belongs inside the performance and safety assurance loop rather than outside it.

Three monitoring principles emerge from the case study. First, threshold-local diagnostics should be linked to the policy itself; $\mathrm{TLCE}(\tau^*)$ and $\mathrm{AOG}(\tau^*)$ are more informative than one global calibration score. Second, fixed-threshold safety and threshold-range robustness should be measured separately, using $R(\tau^*)$ and mean $R(T)$. Third, monthly or rolling audits should be performed whenever regime prevalence changes materially, because a calibration map that looks good on its fitting window can later lead to over-alerting.

For operations, the implication is pragmatic. If the policy threshold is fixed and closely controlled, the main quantity is $R(\tau^*)$. If thresholds vary across units or over time, threshold-averaged risk and S_R become equally important. If the regime changes regularly, monthly or rolling recalibration audits become part of the deployment assurance case. Thus, the airport case study is less about declaring one calibrator universally best than about showing how calibration should be governed under drift.

This study also has explicit boundary conditions. The decision unit is monthly; the tool is designed for sustained regime alerting and cannot detect daily or intra-month disruptions. The feature set uses historical BTS variables only; no weather forecasts, airport-capacity indicators, air-traffic-flow-management, network-state information, or schedule-recovery variables are included, so the model should be read as a leakage-free historical benchmark rather than a complete operational predictor. Risk is unweighted per airport-carrier-month; this tests consistency of the alert policy across decision units, but it does not measure the number of flights or passengers affected. The 5:1 false-negative/false-positive ratio is a stylized asymmetric scenario rather than a validated safety-cost model, although Section VI-C tests alternative ratios. Finally, both the U.S. BTS scope and the 25% high-delay definition are context-specific; transferring the system to other regions, airports, or governance regimes would require re-estimating the threshold, costs, features, and audit envelope. These limits define the assurance boundary within which the reported evidence should be interpreted.

A. Scalability and operational integration

The proposed architecture scales with the number of airport-carrier-month decision units after aggregation. Feature construction is based on group-wise lags and rolling statistics along each airport-carrier time series, so it can be partitioned by airport, carrier, or pair. Inference is a vectorized monthly scoring pass, and the threshold sweep has cost proportional to $|D||T|$, where D is the deployment set and T is the threshold grid. The main scalability bottlenecks are therefore data refresh, retraining frequency, and integration of exogenous forecasts, not the local audit metrics themselves. The present implementation is a monthly batch-assurance architecture rather than a sub-daily streaming system; larger national or multi-region panels would require distributed feature generation, parallel model fitting, and external validation of the same assurance envelope.

Security in this context is broader than the BTS security-delay feature: the protected assets include data feeds, model and calibrator artifacts, threshold configuration, audit logs, and human-override procedures. Deployment should therefore include provenance checks, access control, integrity and anomaly monitoring, signed model/calibration versions, tamper-evident logs, rollback capability, incident response, and human authorization before mitigation is triggered. These controls align the monitoring loop with National Institute of Standards and Technology (NIST) AI risk management, International Civil Aviation Organization (ICAO) aviation-cybersecurity strategy, and European Union Aviation Safety Agency (EASA) airworthiness information-security guidance [18]–[20]. The present paper evaluates predictive and decision-assurance behavior; adversarial robustness, data-poisoning tests, and formal cybersecurity assurance remain deployment prerequisites outside the empirical scope.

VIII. Deployment Scenarios and Monitoring Checklist

The empirical results suggest three recurring deployment scenarios. In a *fixed-threshold governance* scenario, the threshold is centrally defined and rarely changed. In that case, the main assurance quantity is $R(\tau^*)$ together with local diagnostics around τ^*. In a *flexible-threshold governance* scenario, local units or operators may shift the operating point as traffic conditions change. There the pair $\big(R(\tau^*),\ \mathrm{mean}\ R(T)\big)$ becomes more informative than either quantity alone because the same calibration map may improve one and degrade the other. In a *drift-prone seasonal* scenario, prevalence and operating conditions move enough that even a well-fitted calibration map becomes stale. The monthly audit in Figure 5 is a concrete example of this third regime.

Table VII converts the findings into a compact deployment-assurance checklist. The entries are grounded in the observed behaviors of the case study. A jump in action rate without a comparable rise in prevalence indicates potential over-alerting. A strong improvement on the calibration window combined with worse H2 local error indicates calibration-transfer failure. A disagreement between nominal-threshold risk and threshold-averaged risk signals a policy-threshold mismatch rather than a simple model-quality difference. These patterns are not unique to airport operations; they are generic warning signs for AI-based alerting tools that operate under drift.

A second governance issue concerns the gap between the nominal Bayes threshold τ^* and the empirically minimizing threshold $\tau_{\min}$ over the tested range. The empirical threshold

TABLE VII. DEPLOYMENT-ASSURANCE CHECKLIST DERIVED FROM THE CASE STUDY.

Observed signal	Illustration in this study	Recommended assurance response
Action rate rises while prevalence stays low	XGB + Platt in October 2024: action 0.452 at prevalence 0.057	Review the alert threshold or freeze the calibration map until local fit is re-validated
Calibration improves on H1 but worsens on H2	XGB Platt: TLCE 0.013 on H1 versus 0.044 on H2	Treat the calibrator as a time-sensitive component and re-audit it on recent data
Nominal-threshold and threshold-range conclusions disagree	RF + Platt at 5:1: higher $R(\tau^*)$ but lower mean $R(T)$	Report both policy-specific safety and threshold-range robustness before deployment changes

sweep shows that raw XGBoost stays relatively close to the nominal threshold for all three cost ratios, while Platt scaling shifts $\tau_{\min}$ upward: from 0.30 to 0.33 for 2:1, from 0.15 to 0.22 for 5:1, and from 0.06 to 0.13 for 10:1. Random forest shows the same directional effect. This matters operationally because a calibration map can silently change the threshold that an operator would find most effective in practice, even when the nominal cost model is unchanged.

A practical monitoring loop follows naturally. Each audit cycle should log regime prevalence, action rate, $\mathrm{TLCE}(\tau^*)$, $\mathrm{AOG}(\tau^*)$, and both fixed-threshold and threshold-range risk. If these quantities leave the envelope observed on the calibration window, the system should be recalibrated, reverted to a conservative baseline policy, or escalated for human review. For operators and safety owners, this translates statistical evaluation into visible signals: whether alerts are firing too often, whether missed high-delay regimes are increasing, and whether the current threshold still corresponds to the intended policy. A credible model card for AI-based decision-support tools should therefore report not only discrimination metrics, but also the operating threshold, local calibration around that threshold, threshold-range robustness, and at least one temporal audit under changing prevalence.

External transfer should be expressed carefully. The present evidence concerns monthly airport-delay regime alerting, and thresholds, cost ratios, and seasonal mechanisms differ across domains. What transfers is not the operating point itself but the evaluation logic: define the decision unit, prevent leakage, validate chronologically, check calibration locally at the action threshold, and monitor intervention behavior over time.

IX. CONCLUSION AND FUTURE WORK

This paper reframed airport delay regime forecasting as a performance-safety-robustness problem for an AI-based alerting tool operating under temporal drift. It established a leakage-free one-month-ahead protocol, separated predictive performance from nominal-threshold safety and threshold-range robustness, compared model and calibration families under temporal transfer, and used monthly audit to show that calibration effects can reverse across deployment sub-regimes.

Future work should target dynamic recalibration, drift detection, weather and network-state predictors, daily or flight-level units, recurrent and spatio-temporal baselines, flight- and passenger-weighted risk, stakeholder-validated costs, non-U.S. validation, cybersecurity stress tests, and human-supervised threshold governance.

REFERENCES

[1] D. J. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve", *Machine Learning*, vol. 77, no. 1, pp. 103–123, 2009.

[2] A. H. Murphy, "A new vector partition of the probability score", *Journal of Applied Meteorology*, vol. 12, no. 4, pp. 595–600, 1973.

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

[4] J. C. Platt, "Probabilities for SV machines", in *Advances in Large-Margin Classifiers*, A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., Cambridge, MA, USA: MIT Press, 2000, pp. 61–74.

[5] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates", in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 694–699.

[6] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning", in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 625–632.

[7] M. Kull, T. Silva Filho, and P. A. Flach, "Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers", in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 623–631.

[8] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks", in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 1321–1330.

[9] J. Vaicenavicius et al., "Evaluating model calibration in classification", in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, K. Chaudhuri and M. Sugiyama, Eds., ser. Proceedings of Machine Learning Research, vol. 89, PMLR, 2019, pp. 3459–3467.

[10] Y. Ovadia et al., "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift", in *Advances in Neural Information Processing Systems 32*, 2019, pp. 13 991–14 002.

[11] S. Wandelt, X. Chen, and X. Sun, "Flight delay prediction: A dissecting review of recent studies using machine learning", *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 4, pp. 4283–4297, 2025.

[12] N. McCarthy, M. Karzand, and F. Lécué, "Amsterdam to dublin eventually delayed? LSTM and transfer learning for predicting delays of low cost airlines", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9541–9546.

[13] Q. Li, X. Guan, and J. Liu, "A CNN-LSTM framework for flight delay prediction", *Expert Systems with Applications*, vol. 227, p. 120 287, 2023.

[14] U.S. Bureau of Transportation Statistics, *Airline on-time statistics and delay causes*, U.S. Department of Transportation, 2026.

[15] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance", *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, 15:1–15:21, 2012.

[16] L. Breiman, "Random forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[18] National Institute of Standards and Technology, "Artificial intelligence risk management framework (AI RMF 1.0)", National Institute of Standards and Technology, Tech. Rep. NIST AI 100-1, 2023.

[19] International Civil Aviation Organization, *Aviation cybersecurity strategy*, International Civil Aviation Organization, 2022.

[20] European Union Aviation Safety Agency, *AMC 20-42: Airworthiness information security risk assessment*, European Union Aviation Safety Agency, 2023.