

Progressive Aircraft Category Identification from OpenSky ADS-B Trajectories

Prissha Chawla,^{*,1} Manish Kumar,¹ Manuel Arias Chao,² and Chetan S. Kulkarni³

¹University of Cincinnati, Cincinnati, United States

²School of Engineering, Zurich University of Applied Sciences, Switzerland

³KBR Inc., NASA Ames Research Center, United States

*Corresponding author: chawlaps@mail.uc.edu

Abstract

Characterizing an unknown aerial object from motion alone can provide an early evidence layer for broader situational- and intent-assessment systems, especially when imagery, aircraft dimensions, or reliable identity metadata are unavailable. This work studies whether broad aircraft categories can be identified progressively from OpenSky Automatic Dependent Surveillance–Broadcast (ADS-B) trajectories using only causal kinematic information. We consider three classes—helicopter, jet, and propeller—and generate a new class probability vector at every eligible observation as the observed trajectory prefix grows. The benchmark is built from a flight-balanced OpenSky cohort containing approximately 4.21 million observations and 30,276 flights collected from January 1 to May 2, 2023 over an Ohio–Kentucky–Indiana region. Aircraft identities, absolute latitude/longitude, and category metadata are excluded from predictive inputs, and all retained flights from one aircraft remain in a single data partition. Linear SVM, XGBoost, recurrent sequence models, a causal Transformer, and a stacked probability ensemble are evaluated using conventional classification and calibration measures together with flight-centric measures of first correctness, prediction switching, short-term stability, and long-term correct stability. On 5,102 test flights from 255 aircraft excluded from all model-development partitions, the strongest models achieve approximately 95% macro-F1. The stacked ensemble reaches 95.33% macro-F1, while XGBoost reaches 95.13% with the lowest log loss (0.155) and ECE (0.007). Among flights that become correct at least once, the median first-correct prediction for the strongest models occurs at the first eligible position, corresponding to approximately 40 seconds of retained trajectory history; roughly 94–95% of all test flights eventually reach a correct prediction that remains unchanged through the end of the retained trajectory. The results show that aircraft category can often be inferred early from kinematics alone, while models with similar aggregate accuracy can differ materially in probability quality and temporal stability.

Keywords: aircraft classification; ADS-B; OpenSky Network; progressive prediction; trajectory analysis

Abbreviations: ADS-B: Automatic Dependent Surveillance–Broadcast, ECE: Expected Calibration Error, GRU: Gated Recurrent Unit, SVM: Support Vector Machine

1. Introduction

Airspace monitoring increasingly involves a mixture of commercial aircraft, general aviation, rotorcraft, and emerging unmanned systems. In a high-uncertainty setting, a surveillance system may observe motion before it has trustworthy information about the object's identity, mission, or intent. Aircraft category is not itself a threat label, but it can provide a useful first evidence layer: the interpretation of speed, altitude, maneuvering, geographic context, and later anomalous behavior depends strongly on what kind of vehicle is being observed.

We deliberately isolate a first question relevant to broader surveillance and situational assessment: *can a broad aircraft category be inferred from observed kinematics alone, without imagery, physical*

dimensions, or identity-based predictive features? OpenSky ADS-B trajectories provide a large real-world test bed for this question. The longer-term motivation is sensor-agnostic: radar can also provide position and motion histories, so a representation centered on kinematics may transfer more naturally than one tied to self-reported identity fields or visual appearance.

The paper focuses on three broad aircraft categories—HELICOPTER, JET, and PROPELLER—and on a progressive inference setting. Rather than waiting for a complete trajectory, a model produces a class probability distribution repeatedly as new observations arrive. This changes the evaluation question from only “Was the final classification correct?” to “How early did the model become correct, did it stay correct, how often did it switch, and how did confidence evolve?”

The contributions of this work are fourfold:

- a causal progressive aircraft-category classification formulation in which a new prediction is produced for every eligible ADS-B observation;
- an aircraft-disjoint benchmark on real OpenSky trajectories, with absolute geography and aircraft identity excluded from predictive inputs;
- flight-centric evaluation measures that distinguish first correctness, prediction switching, short-term stability, and a correct prediction that remains stable through the rest of the flight; and
- a comparison of engineered-feature and sequence-model approaches showing that greater model complexity does not necessarily provide a substantial practical advantage over a calibrated, interpretable XGBoost model.

The intended scope is deliberately limited. The method does not infer hostile intent, authorization status, or threat. It estimates broad aircraft category from motion and provides a progressively updated probability distribution that can later be combined with geographic, airspace, mission, and anomaly evidence in a broader assessment system. The contribution is therefore not a new classifier architecture; it is the causal progressive formulation, leakage-resistant benchmark design, and evaluation protocol for measuring when a decision becomes correct, stable, and probabilistically trustworthy.

2. Related Work

The OpenSky Network was developed to provide the research community with large-scale access to real-world air-traffic surveillance data, particularly ADS-B and Mode S observations [1]. OpenSky data have since supported work on trajectory prediction, traffic analysis, operational-event detection, anomaly detection, and aircraft characterization. Reviews of four-dimensional trajectory prediction highlight the broad use of surveillance trajectories in data-driven aviation research [2], while recent work has also emphasized the importance of preprocessing noisy and uncertain crowdsourced ADS-B data [3].

A closely related problem is inferring aircraft metadata from behavior. Classi-Fly demonstrated that aircraft categories can be estimated from open trajectory data using behavior-derived features, reporting that movement patterns contain substantial category information even when self-reported metadata are unreliable [4]. Bolton *et al.* used multivariate LSTM–fully convolutional networks for aircraft engine-type classification from public ADS-B data and showed the importance of feature selection and data quality [5]. Other work has explored unsupervised aircraft categorization from motion features [6]. Together, these studies establish that aircraft motion is informative about aircraft type or operating category.

Our focus is complementary but different. Prior behavior-based classification studies primarily ask whether an aircraft can be categorized from a trajectory or collection of trajectories. We ask how

the prediction behaves *while the trajectory is still unfolding*. The model is re-evaluated at every eligible observation using only a causal prefix, and the evaluation is performed at the flight level to quantify correctness onset, switching, stability, and confidence evolution. This distinction matters in operational settings because two models with nearly identical aggregate accuracy can behave differently during the first minutes of observation.

OpenSky trajectories have also been used for anomaly and event detection. Olive et al. compared rule-based and data-driven approaches for recognizing operational events in aircraft trajectories [7]; Corrado et al. studied spatial and energy anomalies in ADS-B trajectories [8]; and Baege et al. demonstrated scalable timing-based anomaly detection on real OpenSky data [9]. These efforts motivate the broader context of this paper: category inference is one component that can eventually inform how unusual motion is interpreted, but category classification and anomaly or intent assessment should remain separate tasks.

3. Data and Experimental Design

3.1 OpenSky study region and source data

The study uses OpenSky historical state vectors over approximately 38°–42° N and 86°–82° W, covering Ohio and portions of Kentucky and Indiana. Hourly Trino queries retained latitude, longitude, barometric altitude, ground speed, heading, callsign, and ground status; sampled timestamps satisfying $\text{time} \% 10 = 0$; removed stale states with $\text{time} - \text{lastcontact} > 15$ seconds; and required at least 30 sampled observations for an aircraft within an hourly query. The archived study snapshot contains 2,912 hourly partitions from January 1 through May 2, 2023. After metadata labeling, flight construction, target-class filtering, and benchmark cohort selection, the analysis snapshot contains 4,211,511 observations from 1,776 aircraft and 30,276 flight records. The 10-second query cadence is reflected in a 10-second median within-flight interval after cleaning.

Supervised category labels were established before model fitting by joining telemetry to OpenSky aircraft metadata on normalized icao24, after deduplicating the metadata to one row per aircraft. The labeling script uses icaoaircrafttype, typecode, manufacturername, and model in a fixed precedence order. It first identifies non-target UAV, glider, balloon/LTA, and selected military records. Among the remaining aircraft, an icaoaircrafttype beginning with H or typecode=UHEL maps to HELICOPTER; an icaoaircrafttype ending in J, or a type code containing B7, A3, GLF, LJ, or CL, maps to JET; and an icaoaircrafttype ending in P or T maps to PROPELLER. Other records map to UNKNOWN. Only the three target labels are retained. The complete executable mapping, including preceding non-target rules, is supplied with the reproducibility material. Thus metadata establishes the supervised target but is not a predictive input.

The benchmark cohort is intentionally balanced by flight count before the final aircraft-disjoint split. After removing flights shorter than 30 observations, all eligible helicopter flights are retained and jet and propeller flights are selected with a seeded aircraft-first procedure until each class matches the helicopter count (10,092 flights per class in the analysis snapshot). This design reduces class-frequency dominance for model comparison but does not represent real-world class prevalence. Predictive inputs exclude icao24, callsign, category metadata, and absolute latitude/longitude. Relative displacement from the start of the retained airborne segment remains available as a causal kinematic feature.

3.2 Flight construction and causal cleaning

Continuous records from the same aircraft are first segmented into upstream flights whenever the observation gap exceeds 600 seconds. Each upstream flight is then ordered by timestamp, duplicate timestamps are removed, and required numeric state fields are forward-filled; no backward fill or

interpolation of numeric kinematics from future rows is used. Airborne-segment construction is an offline preprocessing step. To avoid fragmenting an otherwise continuous segment because of brief status glitches, an internal ground-status run of at most two observations is repaired when it is bounded on both sides by airborne observations. That specific status repair inspects a following observation and therefore should not be interpreted as strictly online preprocessing. The longest continuous airborne run is then retained, and a flight must contain at least 30 airborne observations to remain in the study. After the retained segment is fixed, every engineered feature and every model prediction at position t uses only observations at or before t . After this cleaning, 30,160 flights from 1,774 aircraft and 4,173,175 airborne observations remain.

Predictions begin at airborne position 5. Positions 1–4 are preserved as history, but they are not individually scored because the engineered representation uses a five-observation rolling window. At the nominal 10-second cadence, position 5 corresponds to approximately 40 seconds after the first retained observation.

3.3 Aircraft-disjoint data partitions

All retained flights from a single aircraft identity (icao24) are assigned to one partition. This prevents observations or flights from the same physical aircraft from appearing in both training and test data. The partition roles are separated so that model fitting, tuning, probability calibration, ensemble fitting, and final testing use disjoint aircraft sets. Table 1 summarizes the final partitions. “Eligible observations” exclude positions 1–4 from each retained flight.

Table 1. Aircraft-disjoint data partitions.

Partition	Aircraft	Flights	Eligible observations
Train	1,268	19,877	2,377,090
Tune	151	3,147	501,321
Calibration	50	1,009	187,304
Ensemble development	50	1,025	182,307
Test	255	5,102	804,513

Within the test partition, the 804,513 eligible observations comprise 224,431 helicopter, 232,180 jet, and 347,902 propeller observations from 1,636, 1,753, and 1,713 flights, respectively. During fitting, tuning, calibration, and ensemble development, sample weights give flights equal total weight within each class and balance the classes. The headline test metrics reported below are unweighted over eligible observations; separate flight-centric metrics prevent long flights from being the only view of performance.

4. Progressive Classification Method

4.1 Problem formulation

For flight i , let $\mathbf{x}_{i,1:T_i}$ denote the retained airborne sequence and $y_i \in \{1, 2, 3\}$ the aircraft category. At each eligible position $t \geq 5$, the classifier receives only the causal prefix $\mathbf{x}_{i,1:t}$ and produces

$$\mathbf{p}_{i,t} = f(\mathbf{x}_{i,1:t}), \quad \hat{y}_{i,t} = \arg \max_c p_{i,t,c}, \quad (1)$$

where $\mathbf{p}_{i,t}$ is a three-class probability vector. No observation after t is available to the model or to any derived feature at that point.

The engineered-feature models use a causal transformation $\phi(\mathbf{x}_{i,1:t})$ of the prefix. The sequence models instead receive the instantaneous feature sequence up to t and learn the temporal representation directly.

4.2 Causal feature representations

XGBoost and the Linear SVM use 50 engineered features. The features are grouped in Table 2. They include instantaneous motion, cumulative behavior, running statistics, five-observation rolling variability, and behavioral indicators. Relative east/north displacement is measured from the first retained airborne point rather than from an absolute geographic coordinate.

Table 2. Feature groups used by the engineered-feature models. All quantities are causal.

Group	Examples
Instantaneous motion	Relative east/north displacement, altitude, speed, heading sine/cosine, time delta, speed change, acceleration, altitude change, climb rate, heading change, turn rate, step distance
Cumulative behavior	Radial distance, cumulative distance, cumulative absolute altitude/heading/speed change, net altitude change, elapsed time
Running statistics	Expanding speed and altitude mean/std/min/max, running maximum altitude
Recent variability	Five-observation speed, climb, turn, altitude, and distance statistics; interquartile ranges
Behavioral indicators	Hover flag/count, high-variability flag/count, composite behavior score

The GRU, LSTM, and Transformer use a smaller 16-dimensional instantaneous representation: relative displacement, altitude, speed, heading sine/cosine, time delta, first differences, acceleration, climb rate, turn rate, step distance, elapsed time, and airborne position. They do not receive the cumulative, rolling-window, hover, or composite behavioral features used by XGBoost and the SVM.

All preprocessing parameters are fitted using training data only. Numeric features are median-imputed when necessary and clipped to the 0.001 and 0.999 training quantiles. XGBoost uses the clipped features directly, whereas the SVM and sequence models use standardized features.

4.3 Model families

The principal standalone models evaluated in the study are:

- **Linear SVM:** a linear margin-based baseline using the 50 engineered features. The regularization parameter is tuned on the dedicated tuning partition and the output is probability calibrated.
- **XGBoost:** gradient-boosted decision trees using the 50 engineered features, allowing nonlinear interactions among interpretable kinematic and behavioral quantities [10]. The final run uses learning rate 0.04, maximum depth 8, subsampling and column sampling of 0.85, and early stopping on tuning-set macro-F1; the selected best iteration is 106.
- **GRU:** a two-layer gated recurrent network with hidden size 192 that processes the available instantaneous sequence prefix [11].
- **Causal Transformer:** a four-layer attention model ($d_{model} = 128$, eight heads) using a causal mask so a prediction cannot attend to future observations [12].
- **Stacked probability ensemble:** multinomial logistic regression trained on a reserved ensemble-development partition. For full transparency, the stacker uses 18 probability inputs from six cal-

ibrated base learners: XGBoost, Linear SVM, LSTM, GRU, causal Transformer, and a GRU fusion model that combines sequence and engineered features. Standalone LSTM and GRU-fusion results are included in Appendix 1.

The sequence models use dropout and early stopping on the tuning partition. XGBoost and the deep models are temperature calibrated using only the calibration partition; temperature scaling is a standard post-hoc approach for improving probabilistic calibration [13]. The XGBoost temperature in the final run is 0.8602. The stacked ensemble is then fitted on the separate ensemble-development partition, avoiding the reuse of test data for model combination.

4.4 Use of AI

OpenAI ChatGPT was used as an interactive assistance tool during portions of software debugging, documentation and reproducibility-package auditing, preparation of manuscript figures from author-verified result tables, and manuscript language/organization. It was not used to collect OpenSky observations or to generate the reported trained-model outputs. The authors reviewed and verified the executed analyses, numerical results, methodological descriptions, figures, and scientific interpretation.

5. Evaluation

5.1 Observation-level metrics

We report unweighted test-set accuracy, balanced accuracy, macro-F1, multiclass log loss, and expected calibration error (ECE). Balanced accuracy and macro-F1 reduce the influence of class imbalance by giving each class equal importance. Log loss evaluates the complete probability distribution and penalizes confident wrong predictions more strongly than low-confidence errors. ECE is computed with 15 confidence bins and measures the gap between predicted confidence and empirical accuracy.

Because successive predictions from the same flight are dependent, headline observation-level metrics are supplemented with flight-centric measures rather than treated as independent samples.

5.2 Flight-centric progressive metrics

For each test flight, we evaluate the ordered prediction sequence beginning at airborne position 5.

First correct prediction. The first eligible position at which $\hat{y}_{i,t} = y_i$:

$$t_i^{\text{FC}} = \min\{t : \hat{y}_{i,t} = y_i\}. \quad (2)$$

If the model is already correct at position 5, elapsed time measured from the first *eligible* prediction is zero; in this paper we therefore report the airborne position and interpret position 5 as approximately 40 seconds after the first retained observation at the nominal cadence.

Prediction switching. For a flight with eligible predictions $t = 5, \dots, T_i$, the switch count is

$$S_i = \sum_{t=6}^{T_i} \mathbb{1}[\hat{y}_{i,t} \neq \hat{y}_{i,t-1}]. \quad (3)$$

A low switch count indicates temporal consistency, although a consistently wrong model can also have few switches.

Short-term stable correct. A short-term stable-correct onset is the first position that begins a run of at least five consecutive correct predictions. This captures a locally stable decision while allowing later destabilization to be measured explicitly.

Long-term stable correct. A long-term stable-correct onset is the first position after which every remaining eligible prediction is correct, with at least five predictions available in the stable run. The corresponding rate is the fraction of test flights for which such an onset exists. This metric distinguishes temporary correctness from a decision that remains dependable as more evidence arrives.

We also report the fraction of short-term stable-correct flights that later destabilize and the confidence at stabilization onset. Separate analysis records stable incorrect runs, because a high-confidence wrong decision can be more operationally consequential than a visibly uncertain one.

The stability-onset quantities are retrospective evaluation descriptors rather than online stopping rules. A short-term onset at position t can only be confirmed after the fifth consecutive correct prediction has been observed, while a long-term stable-correct onset is known only after the remainder of the retained flight has been evaluated.

5.3 Paired flight bootstrap

To compare models while respecting within-flight dependence, uncertainty is estimated using 1,000 paired bootstrap resamples at the flight level with 95% confidence intervals. The same sampled flights are used for both models in each paired comparison. The resampling unit is a flight rather than an observation. Because some unseen test aircraft contribute multiple flights, this bootstrap does not additionally cluster at the aircraft level; its intervals therefore account for within-flight dependence but may be less conservative than an aircraft-level bootstrap. We use it as a paired sensitivity analysis for small model differences rather than as evidence of complete statistical independence among flights.

6. Results

6.1 Observation-level classification and calibration

Table 3 reports the main five model results. All nonlinear and sequence-based approaches achieve approximately 95% macro-F1, while the Linear SVM trails at 91.41%. The stacked ensemble has the highest macro-F1 point estimate (95.33%), only 0.20 percentage points above XGBoost (95.13%). XGBoost, however, has the lowest log loss (0.1548) and lowest ECE (0.0070), indicating substantially better probability quality than the stacked ensemble despite nearly identical classification performance.

Table 3. Unweighted observation-level performance on 804,513 eligible observations from 5,102 test flights belonging to aircraft excluded from all model-development partitions.

Model	Accuracy	Balanced acc.	Macro-F1	Log loss	ECE
Stacked ensemble	95.30%	95.52%	95.33%	0.2879	0.0354
XGBoost	95.12%	95.11%	95.13%	0.1548	0.0070
Transformer	94.96%	95.48%	95.00%	0.1791	0.0114
GRU	94.95%	95.27%	94.99%	0.1702	0.0077
Linear SVM	91.32%	91.65%	91.41%	0.4382	0.0380

The paired flight bootstrap supports the interpretation that the ensemble and XGBoost are very close on headline classification. For macro-F1, XGBoost minus ensemble is -0.20 percentage points with a 95% interval of approximately $[-0.58, 0.17]$, and for accuracy the interval also crosses zero. Balanced accuracy is a small exception: the ensemble exceeds XGBoost by about 0.41 percentage points, with the paired interval narrowly excluding zero. Thus, the ensemble has a modest advantage on balanced recall, but there is no broad statistical separation on the main accuracy and macro-F1 measures.

Figure 1 shows the XGBoost row-normalized confusion matrix. Recall is 94.55% for helicopters, 95.66% for jets, and 95.13% for propeller aircraft. The Transformer attains the highest helicopter and jet recall among the evaluated models (96.70% and 97.73%), but its propeller recall is lower (92.00%), illustrating a class-specific trade-off that is obscured by a single aggregate score.

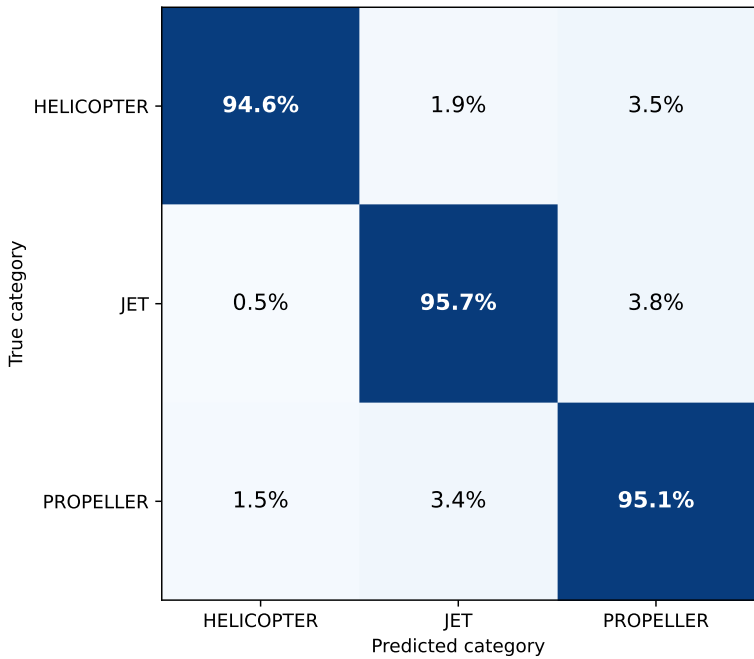


Figure 1. XGBoost row-normalized confusion matrix on the unseen-aircraft test partition. Rows are true categories and sum to 100%; cells show row percentages.

6.2 Progressive accuracy and probability evolution

XGBoost is already strong early in the retained trajectory. In the first 5% trajectory-progress bin, its observation-level accuracy is 93.43% and mean true-class probability is 89.65%. Accuracy rises to roughly 95% as additional evidence accumulates, while the mean true-class probability rises above 92% and remains stable over much of the later trajectory (Figure 2). The progressive contribution is therefore not a claim that category recognition begins from a low-accuracy regime and improves dramatically. Instead, it measures whether an already strong early decision becomes dependable: whether it switches, destabilizes, remains calibrated, or settles permanently on the correct class as evidence accumulates.

6.3 First correctness, switching, and stability

Table 4 summarizes the main flight-centric behavior. The median first-correct position is 5 for every model shown, meaning that among flights that become correct, the median onset occurs at the first

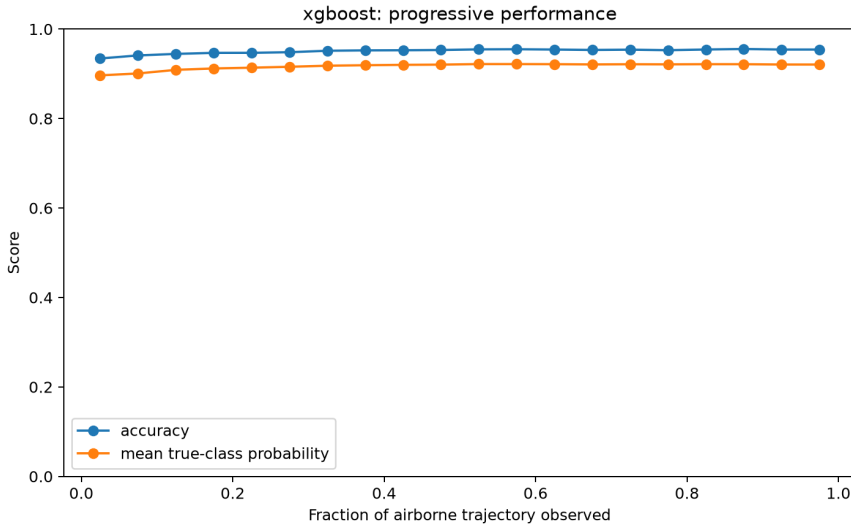


Figure 2. XGBoost accuracy and mean probability assigned to the true class as a function of observed trajectory fraction. Bins are computed over eligible progressive observations.

eligible prediction. With nominal 10-second sampling, position 5 corresponds to approximately 40 seconds after the first retained observation.

Figure 3 shows two deliberately selected, illustrative XGBoost PROPELLER test flights; they are used to make the stability definitions concrete rather than to represent the overall test-set distribution. In panel (a), the prediction is correct from position 5 through flight end: short- and long-term stable-correct onset are both position 5, while short-term stability is confirmed at position 9. In panel (b), a five-correct run also begins at position 5, but one wrong prediction occurs at position 11 ($P_{\text{true}} = 0.300$, top wrong-class confidence = 0.689). Correctness resumes at position 12 and persists to flight end, making position 12 the retrospective long-term stable-correct onset.

The differences appear in what happens *after* that first correct prediction. XGBoost averages 0.58 prediction switches per flight, compared with 0.22 for the Transformer and 0.19 for the GRU; the Linear SVM averages 1.24. Long-term stable-correct rates are approximately 94–95% for the non-linear and temporal models, while the Linear SVM reaches 89.18%. The Transformer has the lowest conditional rate of later destabilization among flights that first reach short-term stable correctness (1.67%); XGBoost is higher at 6.14%, and the Linear SVM is substantially higher at 15.81%. Notably, the Linear SVM is correct at least once on 96.90% of flights despite its much lower long-term rate, illustrating why first correctness alone is an incomplete measure of reliability.

Figure 4 shows the cumulative fraction of XGBoost flights that have reached a first correct prediction and the fraction that have already entered their final permanently correct run as more of the trajectory is observed. By 10% of the retained airborne trajectory, about 80.7% of flights have been correct at least once and about 75.7% have already entered their permanently correct run. The curve then approaches the final long-term stable-correct rate near 95%.

The distinction between confidence and correctness is also important. XGBoost's one-vs-rest calibration curves track the ideal diagonal closely (Figure 5), consistent with its low ECE. Nevertheless, 240 XGBoost test flights (4.70%) contain a final wrong-class run lasting at least five observations through the end of the retained trajectory. At the onset of those long-term stable wrong runs, mean

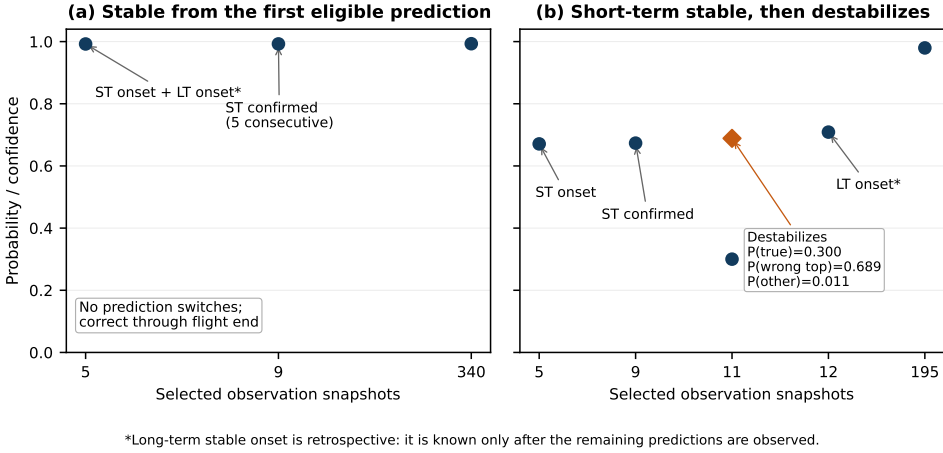


Figure 3. Selected illustrative XGBoost test-flight examples. Both examples are propeller flights chosen to make the progressive stability definitions concrete, not to represent the overall test-set distribution. Navy markers show $P(\text{true class})$ at reported snapshots; the orange diamond is the top wrong-class probability at the destabilizing observation. ST onset is the retrospective start of a five-correct run; LT onset is the retrospective start of the final all-correct run.

Table 4. Flight-centric progressive behavior on 5,102 test flights. Ever correct is a rate over all flights; First pos. is the median among flights that become correct at least once. Destabilized is conditional on first reaching short-term stable correctness.

Model	Ever correct	First pos.	Switches	Long-term	Destabilized
Stacked ensemble	96.69%	5	0.31	94.61%	3.30%
XGBoost	97.26%	5	0.58	94.53%	6.14%
Transformer	95.67%	5	0.22	94.32%	1.67%
GRU	96.55%	5	0.19	94.59%	2.65%
Linear SVM	96.90%	5	1.24	89.18%	15.81%

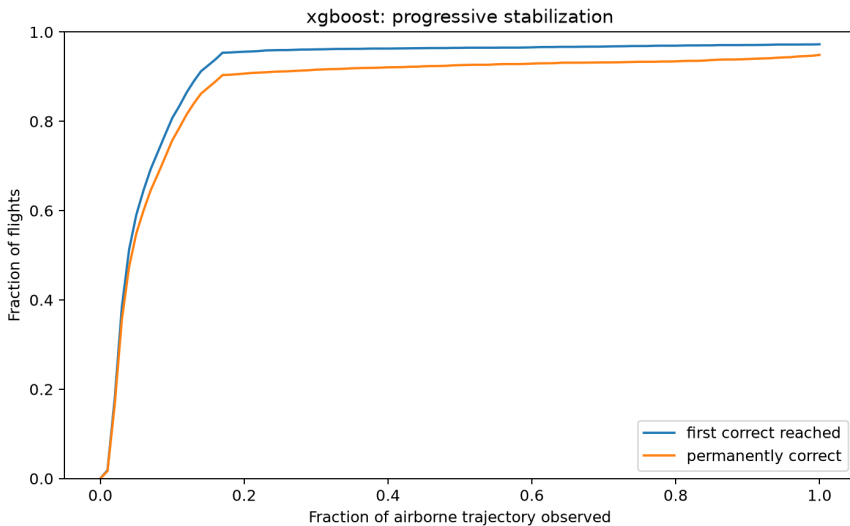


Figure 4. Cumulative XGBoost first-correct and permanently-correct stabilization rates versus observed trajectory fraction.

wrong-class confidence is about 75.1%, and the correct class is still ranked second in 83.3% of cases. High confidence therefore cannot be treated as a substitute for temporal validation and calibration.

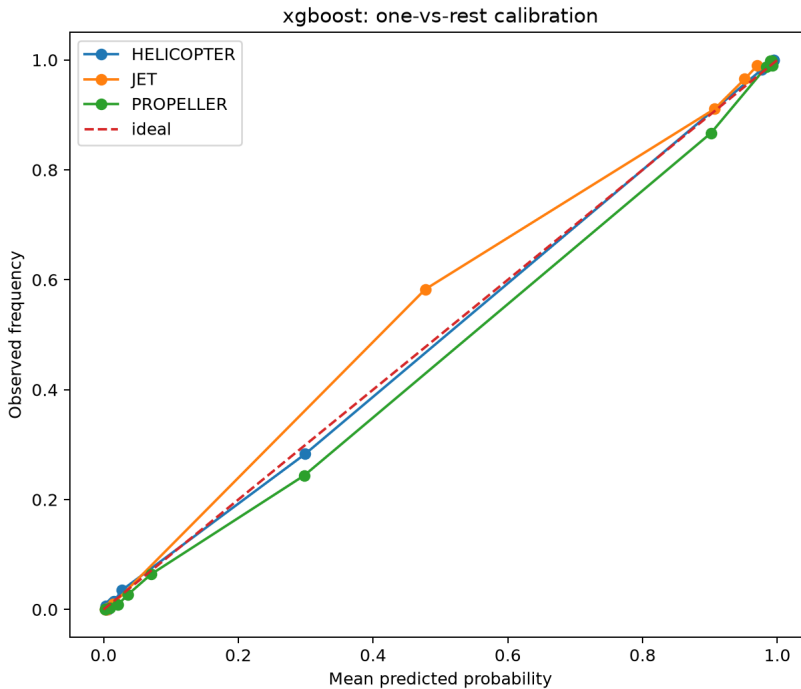


Figure 5. XGBoost one-vs-rest calibration on the unseen-aircraft test partition.

7. Discussion

7.1 What progressive evaluation adds

A conventional test table suggests a simple conclusion: several modern classifiers are all near 95% macro-F1. The progressive analysis reveals a more operationally meaningful picture. Models differ in how often they switch, how often an apparently stable decision later changes, and how reliable their probabilities are. A recurrent model can be temporally smoother than XGBoost even when its overall macro-F1 is nearly identical; the ensemble can have the best balanced accuracy while still being less well calibrated; and a linear model can achieve a correct prediction early on many flights but remain substantially less reliable over the full trajectory.

This distinction is especially relevant for systems intended to characterize an aerial object as observations accumulate. A decision support system may need to expose not only the most likely class, but also whether the prediction has been stable, whether competing classes remain plausible, and whether confidence is calibrated. The progressive framework provides these quantities directly rather than reducing an entire trajectory to one final decision.

7.2 Architecture versus feature representation

The model comparison is a systems-level comparison rather than a controlled architecture-only ablation. XGBoost and the Linear SVM receive 50 engineered causal features, whereas the GRU, LSTM, and Transformer receive a 16-dimensional instantaneous representation and must learn longer-

horizon structure from the sequence itself. Consequently, a performance difference can reflect both model architecture and feature representation. This design is intentional because it compares practical end-to-end alternatives for progressive classification, but it should not be read as evidence that one architecture is intrinsically superior when all models are given identical inputs.

7.3 Why XGBoost is a strong practical baseline

The stacked ensemble has the highest point-estimate macro-F1, but its gain over XGBoost is small and not statistically separated on macro-F1 or overall accuracy in the paired flight bootstrap. XGBoost has the best log loss and ECE, uses one model instead of a six-model probability stack, and operates on explicit physical-motion features such as acceleration, climb rate, turn rate, cumulative distance, and variability. This makes individual predictions amenable to feature-level interpretation and simplifies deployment and maintenance. The sequence models offer lower switching, which may be valuable when temporal smoothness is the dominant requirement, but they do not provide a clear headline classification advantage in this benchmark.

Accordingly, the result is not that one architecture dominates in every dimension. Rather, the study exposes a trade space: the ensemble maximizes aggregate classification, recurrent/attention models reduce temporal switching, and XGBoost provides a favorable combination of near-best classification, strong calibration, interpretability, and implementation simplicity.

7.4 Limitations

Several limitations constrain the interpretation of these results.

First, the supervised category labels are metadata-derived using deterministic rules rather than independently adjudicated airframe labels. The exact mapping is documented and supplied as executable supplementary code, but errors or ambiguities in the underlying OpenSky metadata or in broad type-code heuristics can still become label noise. The three targets are intentionally coarse; in particular, the propeller label combines the metadata rule’s P- and T-suffix aircraft.

Second, the study cohort is deliberately balanced to the helicopter flight count rather than sampled according to operational prevalence. Accuracy, calibration, and class proportions should therefore be interpreted as benchmark results for this cohort, not estimates under real-world traffic priors.

Third, the study does not include a deliberately minimal speed/altitude-only heuristic baseline. The fact that XGBoost already reaches 93.43% accuracy in the first 5% trajectory-progress bin suggests that broad category separability is strong early in many flights. The principal contribution should therefore be interpreted as the progressive reliability benchmark—correctness onset, switching, calibration, and stability—rather than a claim that complex temporal modeling is required to make the classes separable.

Fourth, as discussed above, the model-family comparison also changes feature representation: engineered-feature models receive 50 causal features, while the sequence models receive 16 instantaneous features. The experiment therefore compares practical modeling pipelines, not architecture under strictly identical representations.

Fifth, the paired bootstrap resamples flights. Although the train/test partition is aircraft-disjoint, multiple test flights can come from the same aircraft, so residual within-aircraft dependence is not represented by the bootstrap unit. An aircraft-level resampling analysis would provide a more conservative uncertainty check.

Sixth, the study covers one geographic region and approximately four months in 2023. Receiver coverage, fleet mix, weather, and operational patterns may differ in other regions or seasons, and no separate geographic, seasonal, or radar-domain test is included. Excluding absolute coordinates

reduces direct geographic memorization, but relative displacement, cumulative distance, and maneuver history can still encode route or mission patterns that correlate with class. Crowdsourced ADS-B data also contain noise, missing observations, quantization, and uneven coverage [3]. The preprocessing addresses basic continuity and missing-value issues but does not remove all surveillance uncertainty. Airborne-segment construction includes the bounded status-glitch repair described in Section 3.2, which is an offline cleaning operation rather than a strictly online step.

Finally, the first prediction occurs at position 5, so “early” does not mean zero history: roughly 40 seconds of retained trajectory have already been observed at the nominal cadence. Category inference should also not be interpreted as intent or threat detection. A jet-like or helicopter-like motion pattern is descriptive evidence about vehicle category; higher-level assessment requires additional context such as airspace, geography, expected mission, authorization, and anomalous behavior.

8. Conclusion and Future Work

This paper introduces and evaluates a progressive framework for identifying broad aircraft categories from OpenSky ADS-B kinematics. On 5,102 test flights from aircraft excluded from model development, nonlinear, temporal, and ensemble methods reach approximately 95% macro-F1. Among flights that become correct at least once, the median first-correct prediction for the strongest models occurs at the first eligible point, approximately 40 seconds after the first retained observation. The results also show why aggregate classification alone is incomplete: models with similar macro-F1 differ in probability calibration, prediction switching, destabilization, and long-term stability. XGBoost is a particularly strong practical baseline because it combines near-best classification with the best log loss and ECE and with interpretable causal kinematic features.

The broader motivation is to characterize an aerial object when only motion is initially available. In that setting, aircraft category is one evidence layer, not the final decision. Future work will test whether the causal representation transfers from ADS-B to radar trajectories, extend the category set to additional aircraft and unmanned-system types, and combine category probabilities with geographic, airspace, mission, and behavior context. A natural next step is to treat unusual trajectory behavior as a separate anomaly-evidence channel and study how quickly that evidence becomes reliable. Such extensions move from “what kind of object does this motion resemble?” toward the larger problem of situational and intent assessment while preserving the uncertainty and progressive evidence accumulation emphasized here.

Appendix 1. Auxiliary Standalone Models

The final stacked ensemble uses calibrated probabilities from six base learners, including an LSTM and a GRU-fusion model in addition to the principal standalone models emphasized in the main text. Their standalone results are reported here for transparency.

Appendix 2. Engineered Feature List

The engineered representation contains 50 causal features spanning the five groups in Table 2: instantaneous motion, cumulative behavior, running statistics, recent variability, and behavioral indicators. The exact ordered feature list used by XGBoost and the Linear SVM is implemented directly in the packaged code at `final_pipeline/opensky_compare/features.py`.

Table 5. Complete standalone and ensemble observation-level results.

Model	Accuracy	Balanced acc.	Macro-F1	Log loss	ECE
Stacked ensemble	95.30%	95.52%	95.33%	0.2879	0.0354
XGBoost	95.12%	95.11%	95.13%	0.1548	0.0070
GRU fusion	95.05%	95.33%	95.06%	0.1703	0.0109
Transformer	94.96%	95.48%	95.00%	0.1791	0.0114
GRU	94.95%	95.27%	94.99%	0.1702	0.0077
LSTM	94.75%	94.88%	94.76%	0.1935	0.0205
Linear SVM	91.32%	91.65%	91.41%	0.4382	0.0380

Acknowledgement

The authors acknowledge the OpenSky Network and its contributing sensor operators for making large-scale air-traffic surveillance data available to the research community.

Author contributions

Prissha Chawla: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft. Manish Kumar: Conceptualization, Project Administration, Supervision, Writing – Review and Editing. Manuel Arias Chao: Methodology, Supervision, Writing – Review and Editing. Chetan S. Kulkarni: Conceptualization, Supervision, Writing – Review and Editing.

Open data statement

Source observations and aircraft metadata are available from the OpenSky Network subject to its access conditions and are not redistributed here. The supplement contains the exact query, metadata-labeling and cohort scripts, aircraft split manifests, configuration, tests, and derived result tables.

Reproducibility statement

The configuration-driven Python pipeline covers cleaning, aircraft-disjoint splitting, causal features, training, calibration, prediction, evaluation, and figure generation. The recorded environment uses Python 3.12.3, scikit-learn 1.9.1, NumPy 2.5.3, pandas 3.0.6, and PyTorch 2.14.1, with XGBoost for boosted trees. The supplement includes run scripts and tests; a full numerical rerun additionally requires the OpenSky-derived source Parquet described there.

References

[1] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. “Bringing Up OpenSky: A Large-Scale ADS-B Sensor Network for Research.” *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 2014, pp. 83–94. doi:10.1109/IPSIN.2014.6846743.

[2] Weili Zeng, Xiao Chu, Zhengfeng Xu, Yan Liu, and Zhibin Quan. “Aircraft 4D Trajectory Prediction in Civil Aviation: A Review.” *Aerospace* 9(2), 2022, 91. doi:10.3390/aerospace9020091.

[3] Xavier Olive, Jan Krummen, Benoit Figuet, and Richard Alligier. “Filtering Techniques for ADS-B Trajectory Preprocessing.” *Journal of Open Aviation Science* 2(2), 2025. doi:10.59490/joas.2024.7882.

[4] Martin Strohmeier, Matthew Smith, Vincent Lenders, and Ivan Martinovic. “Classi-Fly: Inferring Aircraft Categories from Open Data.” *ACM Transactions on Intelligent Systems and Technology* 12(6), 2021, pp. 1–23.

doi:10.1145/3480969.

417

- [5] Sarah Bolton, Richard Dill, Michael R. Grimaila, and Douglas Hodson. “ADS-B Classification Using Multivariate Long Short-Term Memory–Fully Convolutional Networks and Data Reduction Techniques.” *The Journal of Supercomputing* 79, 2023, pp. 2281–2307. doi:10.1007/s11227-022-04737-4. 418
419
420
- [6] Nicolas Vincent-Boulay and Catharine Marsden. “A New Approach to Aircraft Categorization Using Machine Learning to Analyze Aircraft Behaviour.” *AIAA AVIATION 2023 Forum*, 2023. doi:10.2514/6.2023-4214. 421
422
423
- [7] Xavier Olive, Junzi Sun, Adrien Lafage, and Luis Basora. “Detecting Events in Aircraft Trajectories: Rule-Based and Data-Driven Approaches.” *Proceedings* 59(1), 2020, 8. doi:10.3390/proceedings2020059008. 424
425
- [8] Samantha J. Corrado, Tejas G. Puranik, Olivia Pinon Fischer, and Dimitri N. Mavris. “A Clustering-Based Quantitative Analysis of the Interdependent Relationship Between Spatial and Energy Anomalies in ADS-B Trajectory Data.” *Transportation Research Part C: Emerging Technologies* 131, 2021, 103331. doi:10.1016/j.trc.2021.103331. 426
427
428
429
- [9] Lukas Baeye, Patrick Schaller, Vincent Lenders, and Martin Strohmeier. “Scaling the Timing-Based Detection of Anomalies in Real-World Aircraft Trajectories.” *Journal of Open Aviation Science* 1(2), 2023. doi:10.59490/joas.2023.7205. 430
431
432
- [10] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System.” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794. doi:10.1145/2939672.2939785. 433
434
435
- [11] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation.” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1724–1734. doi:10.3115/v1/D14-1179. 436
437
438
439
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need.” *Advances in Neural Information Processing Systems* 30, 2017. 440
441
442
- [13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. “On Calibration of Modern Neural Networks.” *Proceedings of the 34th International Conference on Machine Learning* 70, 2017, pp. 1321–1330. 443
444