Scalable Arrival Flight Delay Prediction: Multi-airport Benchmarking of Random Forest, XGBoost and CatBoost

Soufiane Momtaz*, Otmane Idrissi, Abdelmajid Bousselham and Mohammed Mestari

Higher Normal School of Technical Education (ENSET) Mohammedia Hassan II University, Casablanca, Morocco

Abstract

Predicting arrival flight delays is a pressing challenge for air traffic management, particularly in multi-airport scenarios where data is heterogeneous and highly imbalanced. This paper benchmarks three leading tree-based ensemble models: Random Forest, XGBoost, and CatBoost, within a comprehensive evaluation framework that encompasses discrimination, calibration, error distribution, and operational decision support. The results confirm the superiority of boosting techniques in scalability and predictive reliability, with CatBoost showing enhanced handling of categorical variables and XGBoost excelling in ranking-based metrics. By situating the analysis in a multi-airport context, the study advances beyond single-site evaluations, offering a roadmap for deploying robust predictive solutions in real-world aviation networks. These insights not only establish a methodological benchmark but also point to future research directions in hybridization and calibration for operational deployment.

Keywords

Arrival Flight Delay Prediction, Random Forest, XGBoost, CatBoost, Scalability

1. Introduction

Flight delays remain one of the most persistent and costly challenges in aviation. According to the Federal Aviation Administration (FAA), flight delays cost the U.S. economy approximately $33 billion annually, with arrival delays accounting for nearly 60% of passenger-related costs. Eurocontrol estimates a cost in Europe exceeding €17 billion per year. Arrival delays, more than departure delays, directly affect network stability by causing missed connections, crew scheduling conflicts, and airport congestion. The causes of flight delays are diverse, such as weather disturbances, airspace and runway congestion, technical malfunctions, staffing issues, and operational capacity constraints.

Accurate arrival delay prediction can offer significant operational benefits. It enables better scheduling of airport gates, optimized crew management, and efficient fleet rotation, while also reducing unnecessary fuel burn and mitigating CO_2 emissions to protect the environment. Moreover, improving delay forecasts enhances passenger satisfaction by reducing uncertainty, minimizing missed connections, and improving communication, which minimizes passenger's disturbance and discomfort

Because the main causes of flight delays interact across multiple stages of flight operations, predicting arrival delays remains one of the most complex challenges in aviation analytics. Despite the relevance of this problem, systematic multi-airport benchmarking of predictive models is an under discovered fields. Most studies focus on single-airport datasets, limiting generalization.

In this paper, we join the research contributions aimed at evaluating predictive models across heterogeneous aviation environments, where complex traffic patterns and operational constraints differ. This study presents a robust benchmarking of Random Forest, XGBoost, and CatBoost under a unified experimental protocol. Scientifically, it contributes to the comparative analysis of ensemble methods in large-scale aviation data. Operationally, it demonstrates the potential of predictive models to support real-time traffic management, reduce costs, and improve system reliability.

ICAIW 2025: Workshops at the 8th International Conference on Applied Informatics 2025, October 8–11, 2025, Ben Guerir, Morocco

*Corresponding author.

soufiane.momtaz@gmail.com (S. Momtaz); iodrimane@gmail.com (O. Idrissi); bousselham@enset-media.ac.ma (A. Bousselham); mestari@enset-media.ac.ma (M. Mestari)

0009-0004-5650-2488 (S. Momtaz); 0000-0002-5325-5811 (O. Idrissi); 0000-0001-5458-2294 (A. Bousselham); 0000-0002-9828-1861 (M. Mestari)

2. Related Work

The prediction of flight delays is a promising research topic, when systematic multi-airport benchmarking of ensemble methods under a unified protocol is still underexploited, most previous studies either analyze single-airport datasets or evaluate only one algorithmic family, which limits generalization perspectives. This gap underscores the originality and importance of the present study, which provides a scalable comparative evaluation of Random Forest, XGBoost, and CatBoost across multiple airports. Scientifically, this clarifies the relative strengths of widely used ensemble methods and operationally, it provides actionable insights for decision-makers in air traffic management.

Early studies laid the foundation for this domain. Rebollo and Balakrishnan [1] developed a methodology to characterize and predict delay propagation at the U.S. network level, establishing baseline approaches. EUROCONTROL [2] highlighted the massive economic cost of delays in Europe, reinforcing the urgency of predictive solutions.

Subsequent works introduced predictive modeling with both statistical and machine learning approaches. Gopalakrishnan and Balakrishnan [3] compared competing models for airport delay forecasting, while Choi et al. [4] demonstrated the benefits of non-linear ML methods over traditional statistical baselines. Yazdi et al. [5] introduced a feature-rich ML model incorporating attention mechanisms to improve accuracy, and Zhang et al. [6] proposed an APR-LSTM framework that explicitly captured spatio-temporal dependencies. Mamdouh et al. [7] advanced this line of work by integrating multiple categories of features (operational, weather, traffic) in their FDPP-ML model.

Multi-airport and network-level approaches have gained attention in recent years. Wang et al. [8] demonstrated scalable ETA prediction across multi-airport approach areas. Kiliç et al. [9] analyzed systemic delay patterns across the U.S. network, identifying spatial heterogeneity. Hatıpoğlu et al. [10] evaluated predictive modeling of arrivals exceeding 15 minutes at a European hub, while Ajayi et al. [11] introduced network centrality measures to quantify the role of connectivity in delay propagation.

Tree-based ensemble models are particularly well suited for aviation data. Cao et al. [12] showed that integrating multiple operational and environmental features improves ML-based prediction accuracy. Pineda-Jaramillo et al. [13] extended this by incorporating explainable ML, enhancing interpretability. Gao et al. [14] constructed airport time profiles to analyze delay propagation, while Dai et al. [15] proposed the hybrid COWRF model, combining weather forecasting with ML.

Benchmarking studies further emphasize the competitiveness of ensembles. Beltman et al. [16] systematically compared Random Forest, CatBoost, and DNNs for airline delay probability prediction, finding that ensemble models remain strong baselines. Ribeiro et al. [17] applied predictive analytics for airport capacity management, showing the policy implications of delay forecasting. Yuan et al. [18] proposed a multi-attribute ML framework for departure delay prediction, reinforcing the adaptability of ensemble models. Cook and Tanner [19] provided a broad review of performance and delay research, while Xu et al. [20] surveyed ML applications for air traffic management, concluding that ensemble learning remains central to predictive analytics in aviation.

The previous works confirm that boosting models (XGBoost, CatBoost) often outperform Random Forest in predictive accuracy, yet RF remains a valuable baseline for robustness. The lack of large-scale multi-airport benchmarks under unified protocols justifies the contribution of the present study.

3. Material and methods

3.1. Justification of Model Choice and Comparative Analysis

In the context of flight delay prediction, the selection of Random Forest (RF), XGBoost (XGB), and CatBoost (CB) as benchmark algorithms is strongly motivated by their complementary strengths and their proven success on structured tabular datasets in other fields of research.

Taken together, the comparison between RF, XGBoost, and CatBoost allows us to evaluate three different paradigms: a classical bagging ensemble, an optimized gradient boosting algorithm, and a

modern boosting framework specialized for categorical data. This selection as shown at Table 1 ensures both methodological diversity and relevance to the practical challenges of aviation delay prediction.

Table 1
Comparative main characteristics of RF, XGBoost and CatBoost

Criterion	Random Forest (RF)	XGBoost (XGB)	CatBoost (CB)
Type of Ensemble	Bagging (Independent trees)	Optimized Gradient Boosting	Gradient Boosting (Ordered)
Over-fitting Risk	Low	Higher	Reduced
Hyper-parameter Tuning	Low	High	Medium
Training Speed	Fast	Very fast	Medium
Expected Accuracy	Good	Very high	High
Noise Robustness	High	Medium	Medium/High

3.2. Dataset

The large dataset employed in this study is a public domain dataset of Bureau of Transportation Statistics (BTS) covers the period from 2013 to 2023 in the United States, offering a rich and diverse representation of flight arrival operations across a wide temporal horizon. It contains 171,666 records and 72 variables, with a representation of both raw operational indicators and engineered features, with an important score including 21 carriers and 395 airports, this ensures that this dataset is appropriate for our purpose to provide a heterogeneity dynamics and a multiplicity of influencing factors.

The structure of the dataset reflects an important balance between raw identifier sand it can technically help to preserve categorical diversity while maintaining computational efficiency.

3.3. Hardware and software configuration

The experiments were conducted in an optimized environment, designed to balance scalability with robustness for large-scale arrival delay prediction. This setup ensured an efficient, reproducible, and resource-conscious environment for benchmarking Random Forest, XGBoost, and CatBoost across multiple experimental splits.

3.3.1. Hardware configuration

- GPU detection was enabled with automatic fallback to CPU if no GPU was available.
- When GPU resources were available, both XGBoost and CatBoost were executed in GPU.
- The code leveraged QuantileDMatrix for XGBoost, reducing GPU memory usage.
- Data types were compressed to minimize RAM usage.

3.3.2. Software configuration

- Python version: 3.10 (default runtime for most Colab/A100 environments).
- Python scientific stack: NumPy, Pandas, Matplotlib, SciPy, Statsmodels, Seaborn, TQDM.
- Machine learning libraries: scikit-learn (RandomForestRegressor), XGBoost, CatBoost, and Optuna for hyperparameter optimization with pruning strategies (MedianPruner).
- Versions installed via pip: pandas, numpy, scikit-learn, xgboost, catboost, optuna, scipy, scikit-posthocs, seaborn, statsmodels.
- Randomness was controlled with fixed seeds (np.random.seed and random.seed) for reproducibility.

3.3.3. Configuration details

- Results were stored in outputs_sprint/ with optional linkage to Google Drive for minimal local disk usage.

- Hyper-parameter tuning was executed with Optuna for all three models with early stopping.
- Disk writes were explicitly disabled for CatBoost to optimize I/O performance.
- This setup ensured an efficient, reproducible, and resource-conscious environment for benchmarking Random Forest, XGBoost, and CatBoost across multiple experimental splits.

4. Simulation and evaluation

4.1. Model Design and Implementation

The study was designed to provide a fair comparison of Random Forest, XGBoost, and CatBoost for the prediction of arrival flight delays. All models were implemented under a unified pipeline to ensure consistency in feature processing, training-validation splits, and evaluation metrics. The following subsections describe the features of model design and implementation details.

4.1.1. Configuration Experimental Splits

To mimic realistic forecasting conditions, a rolling-origin evaluation strategy was applied. For each test year T, models were trained on all available data up to year T-2, validated on year T-1, and tested on year T. This procedure generated multiple non-overlapping train/validation/test splits spanning the entire temporal coverage of the dataset. Such a design enabled the assessment of robustness and generalization across time while avoiding optimistic bias.

4.1.2. Random Forest Implementation

The Random Forest model was implemented using scikit-learn's Random Forest Regressor. Hyperparameters tuned included the number of estimators, maximum tree depth, maximum feature sampling strategy, and minimum samples per leaf and split. Training was executed in parallel using all available CPU cores. RF served as a baseline due to its robustness and interpretability, though it is less memory-efficient and less adaptive to high-cardinality categorical data compared to boosting methods.

4.1.3. XGBoost Implementation

XGBoost models were trained using the GPU-accelerated gpu_hist algorithm with gpu_predictor when GPUs were available. A memory-efficient QuantileDMatrix was employed to reduce VRAM usage. Hyperparameters tuned included maximum depth, learning rate, regularization terms (L1 and L2), minimum child weight, subsampling and column sampling rates, and maximum bin size. Early stopping with 40 rounds was applied to prevent overfitting. Training iterations were optimized per split, with best iteration counts retained for final model retraining.

4.1.4. CatBoost Implementation

CatBoost regressors were implemented with GPU acceleration (task_type="GPU") where available. Hyperparameters tuned included depth, learning rate, L2 regularization, and bagging temperature. The overfitting detector (od_type="Iter", od_wait=40) was used to automatically determine the best iteration. CatBoost handled categorical features through its native permutation-driven encoding, though in this study features had already been transformed into dummy variables for consistency across models. Disk writes were disabled to reduce I/O load, and seeds were fixed for reproducibility.

4.2. Tuning and training

The process involved two main stages: hyperparameter tuning using Optuna and final model training on optimized configurations. This ensured a fair and reproducible comparison across models while adapting each algorithm to the characteristics of the arrival delay dataset.

Hyperparameter optimization was conducted with the Optuna framework, leveraging its MedianPruner strategy to accelerate convergence by discarding underperforming trials early. Each model was assigned a search space adapted to its algorithmic structure. For Random Forest, the parameters tuned included the number of estimators, tree depth, feature sampling strategy, and minimum split/leaf sizes. For XGBoost, the search covered tree depth, minimum child weight, learning rate (eta), subsampling ratios, regularization terms, and maximum bin size, with GPU acceleration enabled (gpu_hist, gpu_predictor). CatBoost tuning focused on tree depth, learning rate, L2 regularization, and bagging temperature, with GPU computation used when available. Across all models, early stopping was employed to avoid overfitting: 40 rounds for XGBoost and an overfitting detector (od_wait=40) for CatBoost.

After the tuning stage, the best hyperparameters identified by Optuna were selected for each algorithm. Random Forest was trained in parallel mode using all available CPU cores to handle the high-dimensional feature space. XGBoost models were trained with the optimal number of boosting iterations found during tuning, using either standard DMatrix or the memory-efficient QuantileDMatrix depending on GPU support. CatBoost models were trained on Pools with categorical handling and applied the best iteration selected by the built-in validation process. In all cases, compact data representations (float32, uint8) and rolling training-validation splits were applied to control memory usage and mimic operational forecasting settings.

4.3. Evaluation protocol

The evaluation can be considered rigorous and reproducible with a fair competition rules across algorithms and robustness across temporal and operational contexts. This ensures that this comparative benchmarking of the models is not limited only to raw accuracy scores but has an important extension to many other aspects like computational efficiency, robustness, and statistical confidence.

4.3.1. Performance Metrics

Evaluation was based on complementary error and fit metrics. The Mean Absolute Error (MAE) captured the typical deviation between predicted and observed delays, while the Root Mean Squared Error (RMSE) penalized larger errors, reflecting the sensitivity of delay distributions to extreme values. The Coefficient of Determination (R^2) measured the explanatory power of each model relative to a baseline mean predictor. Metrics were computed globally, as well as by subgroup (e.g., by airport and by carrier), to assess both overall accuracy and heterogeneity across operational contexts.

4.3.2. Statistical significance testing

The evaluation protocol incorporated statistical testing to assess the robustness of observed differences. Pairwise Wilcoxon signed-rank tests were applied on grouped metrics (e.g., per airport MAE) to determine whether differences between models were statistically significant. In addition, a Friedman test followed by a Nemenyi post-hoc analysis was performed to compare all models simultaneously across multiple groups, enabling a robust assessment of relative ranking stability.

4.3.3. Residual Analysis and Diagnostics

To further probe model behavior, residual distributions were examined. Histograms and QQ plots were generated to analyze error normality and detect systematic biases. These diagnostics provided insights into whether certain models were prone to underestimating or overestimating delays.

5. Results and discussion

5.1. Global Performance Tests

The Figure 1 presents the global averages of evaluation metrics calculated for each of the models compared, Blue bars correspond to MAE orange bars to RMSE, and green bars to R^2. The vertical axis indicates the numerical values of the metrics.

This figure provides a synthetic view of the average performance of the models across the dataset and test periods. It is observed that Random Forest (RF) achieves the best overall performance, with the lowest MAE and RMSE values as well as a highest R^2. XGBoost (XGB) delivers intermediate results, close to RF but slightly less accurate in terms of absolute and squared errors. CatBoost (CAT) lags behind, showing higher MAE and RMSE values and a lower R^2.

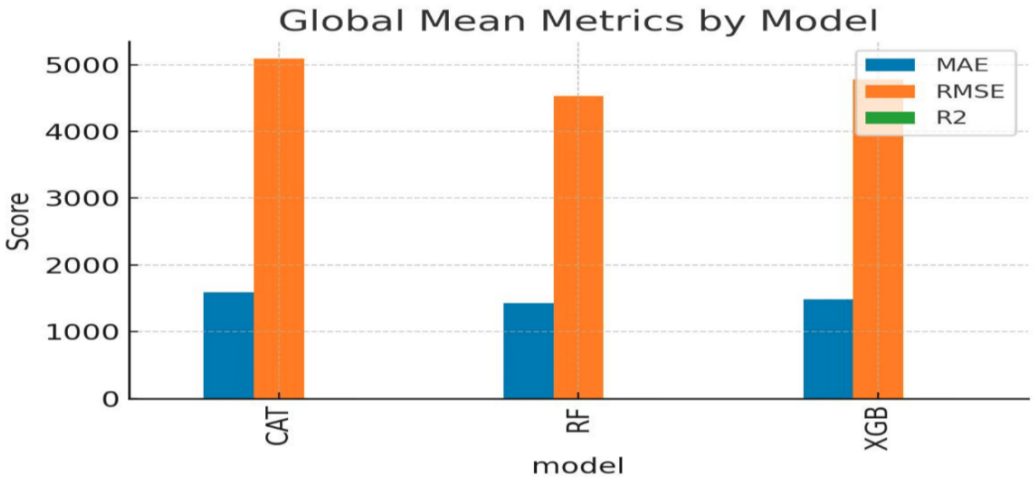

Figure 1: Global Mean Metrics by Model

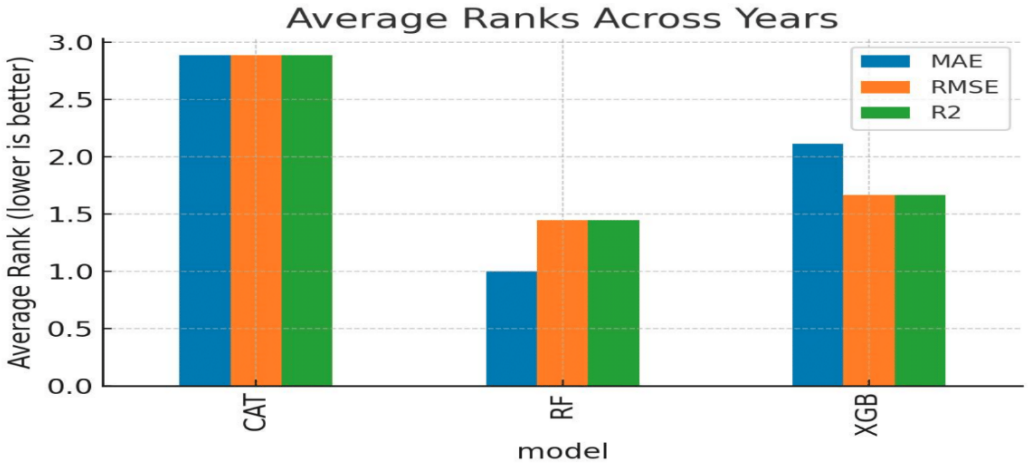

Figure 2: Average Ranks Across Years

The Figure 2 illustrates the average ranks of the three models calculated across multiple years for three evaluation metrics: MAE, RMSE, and R^2. This approach synthesizes multi-year results into a comparative ranking framework, allowing direct evaluation of model consistency and performance.

The results show that Random Forest (RF) consistently achieves the lowest average ranks, especially for MAE, confirming its robust superiority across years. XGBoost (XGB) occupies a middle position, ranking second overall, while CatBoost (CAT) is systematically ranked last across all metrics. These

findings suggest that RF is not only strong in terms of absolute error values but also maintains the most consistent performance across time windows. XGB remains competitive, though slightly less stable.

5.2. Temporal Performance Tests

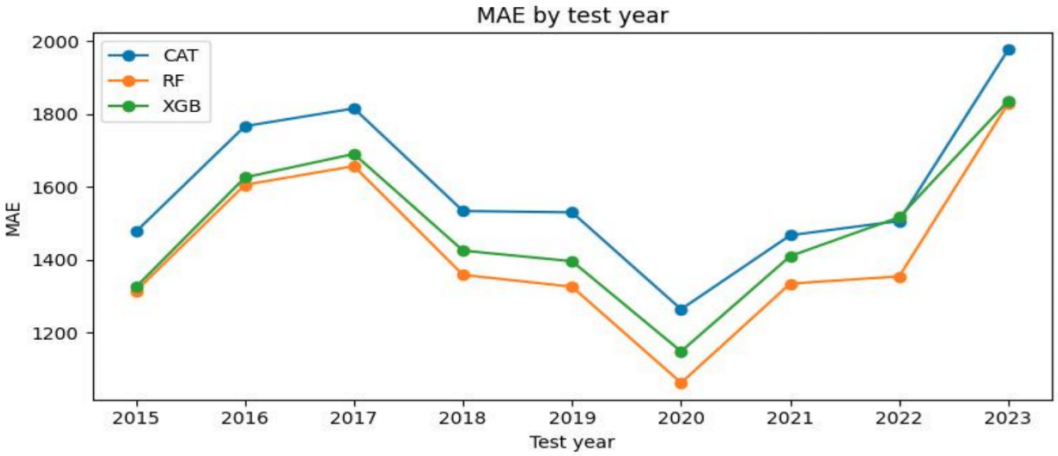

Figure 3: MAE by Test Year

The Figure 3 shows the yearly evolution of MAE for the models over the test years from 2015 to 2023. This temporal analysis highlights how the models' predictive accuracy fluctuates across different periods, providing insights into stability and adaptability.

The results demonstrate that Random Forest (RF) maintains the lowest MAE in most years, particularly between 2018 and 2022, where its performance clearly outperforms both XGBoost and CatBoost. XGB follows closely, often ranking second, while CAT consistently shows higher MAE values, indicating less precise predictions across years. The general upward trend observed in 2023 for all models suggests increased difficulty in prediction during that year, possibly due to external factors such as shifts in traffic patterns or operational disruptions. Overall, this figure reinforces RF's dominance in terms of robustness and accuracy over time, with XGB as a competitive alternative.

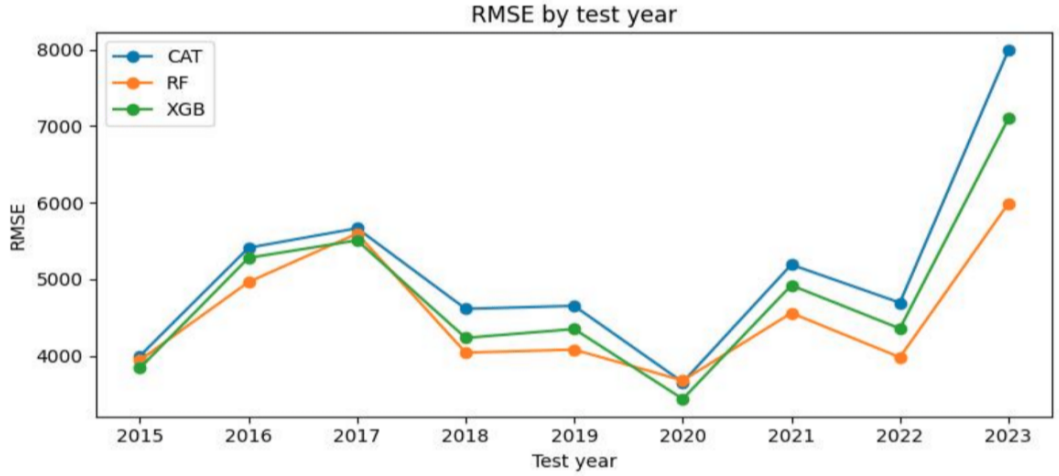

Figure 4: RMSE by Test Year

The Figure 4 shows the yearly evolution of RMSE for the models from 2015 to 2023, RMSE provides a measure of prediction error that penalizes larger deviations, which explains it's value.

The temporal trends reveal that Random Forest (RF) consistently achieves the lowest RMSE values in most years, particularly from 2018 through 2022, underscoring its ability to minimize large errors better than its competitors. XGBoost (XGB) remains close to RF, often ranking second, while CatBoost (CAT) exhibits systematically higher RMSE values, confirming its weaker ability to control error magnitudes. The sharp increase in 2023 across all models suggests a more challenging predictive environment, possibly linked to anomalies or structural changes in the dataset. These results strengthen the conclusion that RF is the most reliable model for controlling large deviations.

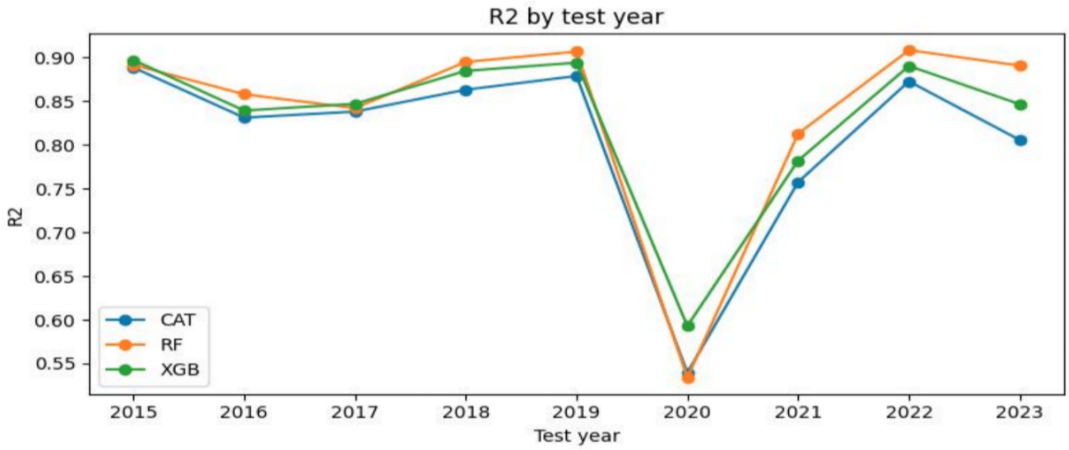

Figure 5: R^2 by Test Year

The Figure 5 presents the yearly evolution of the coefficient of determination (R^2) related to the models from 2015 to 2023. R^2 quantifies the proportion of variance in the target variable explained by the model, with higher values indicating a sign of stronger predictive capability. This temporal view allows an assessment of how well each model captures variability.

The analysis shows that Random Forest (RF) and XGBoost (XGB) achieve consistently higher R^2 values compared to CatBoost (CAT), particularly during stable years such as 2018, 2019, and 2022. But, CAT was far behind, that indicates that it captures less of the underlying variance in delays. The sharp decline in 2020 across all models reflects a disruption in data structure due to the pandemic impact, which reduced model explanatory power.

5.3. Error Distribution Tests

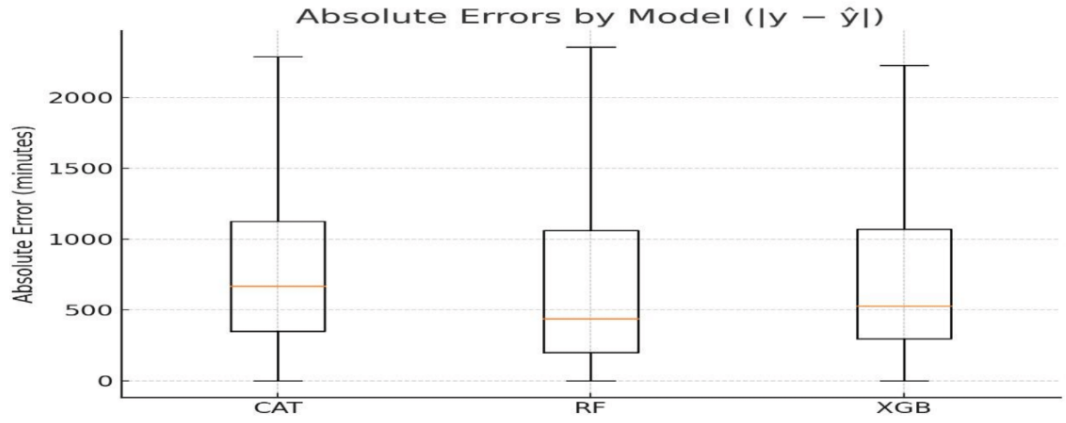

Figure 6: Absolute Errors by Model

The Figure 6 displays boxplots of the absolute prediction errors for the models. The vertical axis

indicates the magnitude of absolute errors in minutes, while the horizontal axis lists the models. Each boxplot summarizes the distribution of errors, with the median shown as the central line, the interquartile range represented by the box, and whiskers extending to capture the spread of values. Outliers beyond the whiskers indicate instances of unusually high errors.

The distribution of errors suggests that Random Forest (RF) yields the lowest median error and a more compact interquartile range, indicating stronger reliability and fewer large deviations compared to the other models. XGBoost (XGB) shows intermediate performance, with a median error slightly higher than RF and broader variability. CatBoost (CAT) exhibits the highest median error and comparable or greater dispersion, underlining its relative weakness in minimizing typical prediction errors. The presence of long whiskers across all models reveals that extreme outliers persist in the dataset, but RF demonstrates a superior ability to constrain error variability.

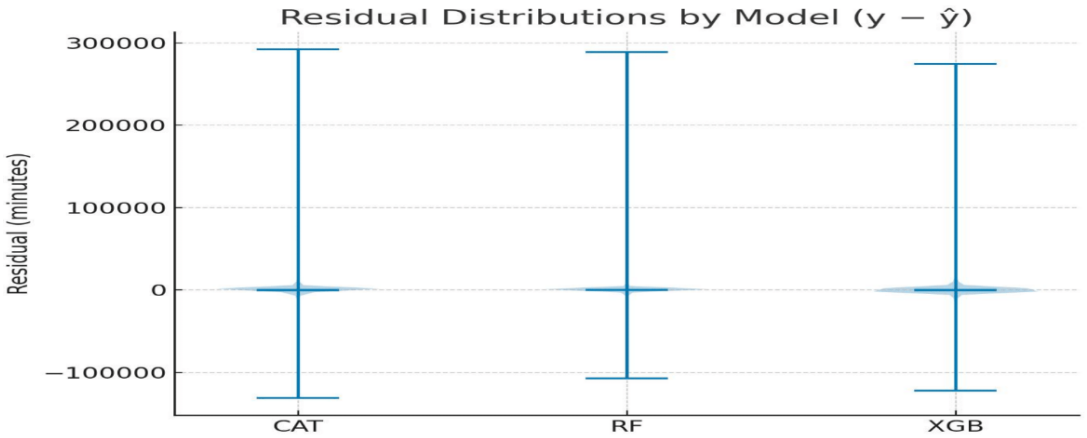

Figure 7: Residual Distributions by Model

The Figure 7 presents violin plots of the residual distributions $(y - \widehat{y})$ for the three models. The vertical axis measures residuals in minutes, while the horizontal axis distinguishes the models. Each violin plot shows the full distribution of residuals: the central thickness reflects the density of values around specific ranges, while the upper and lower extensions capture extreme positive and negative deviations. This highlights both the central concentration of errors and the extent of outliers.

The distributions reveal that all models exhibit highly skewed residuals with heavy tails, reflecting the presence of extreme delays that are difficult to predict accurately. Random Forest (RF) and XGBoost (XGB) display slightly tighter central distributions around zero, suggesting a modest advantage in controlling bias compared to CatBoost (CAT). CAT shows a broader spread and heavier tails, underlining its weaker ability to manage extreme deviations. While none of the models eliminates large outliers, RF and XGB demonstrate stronger stability around the mean.

5.4. Statistical Significance Tests

The Figure 8 shows a heatmap of p-values obtained from pairwise Wilcoxon signed-rank tests comparing the three models based on per-airport Mean Absolute Error (MAE). The rows and columns represent the models being compared, and the color scale encodes the magnitude of the p-values, with numerical values displayed in scientific notation. Lower p-values (darker cells) indicate stronger evidence of statistically significant differences in performance between the models.

The results demonstrates extremely small p-values across all pairwise comparisons, confirming that the differences in MAE between the models are statistically significant at the airport level. Specifically, RF consistently shows significant superiority over both XGB and CAT, while XGB also significantly outperforms CAT. The very low values (on the order of 10^{-54} to 10^{-61}) provide strong evidence that the observed performance gaps are not due to chance. This statistical validation reinforces the conclusion that RF is the most reliable model overall, followed by XGB then by CAT.

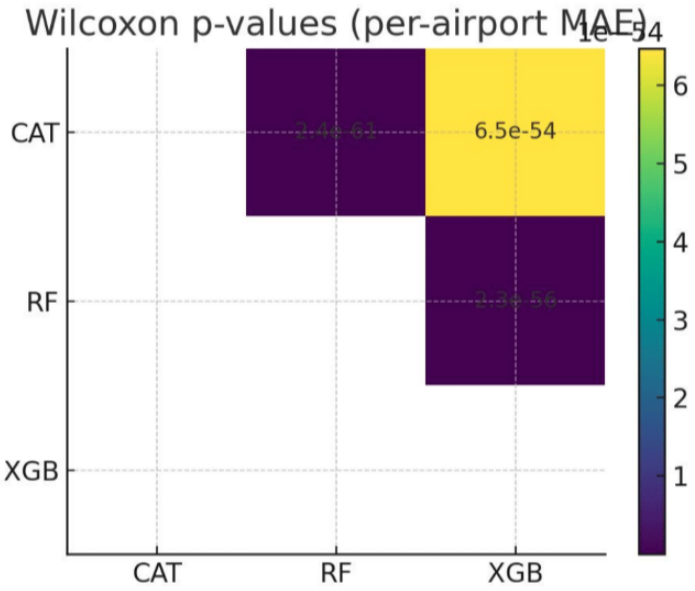

Figure 8: Wilcoxon p-values (per-airport MAE)

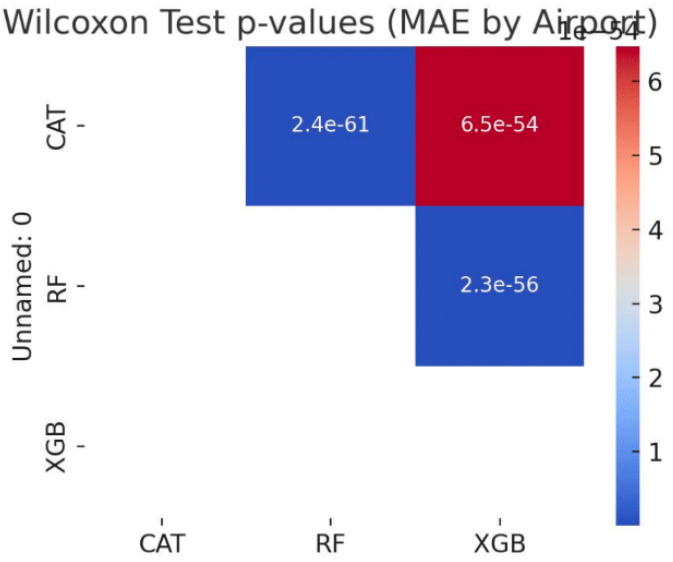

Figure 9: Wilcoxon Test Heatmap (MAE by Airport)

The Figure 9 provides a heatmap visualization of p-values obtained from Wilcoxon signed-rank tests comparing model performance in terms of Mean Absolute Error (MAE) across airports. Rows and columns represent the models, and each cell reports the p-value for the pairwise test. The color scale highlights the magnitude of the p-values, with darker or cooler shades indicating lower values and stronger evidence of significant differences.

The heatmap confirms extremely low p-values, showing that the observed differences in MAE between models are statistically significant at the airport level. Random Forest (RF) is statistically superior to both XGBoost and CatBoost, while XGB also significantly outperforms CAT.

5.5. Operational Tests (Classification: Late $\geq$ 15 min)

The Figure 10 presents the confusion matrix of the Random Forest (RF) model for the classification task of predicting arrival delays. The horizontal axis represents the predicted labels while the vertical

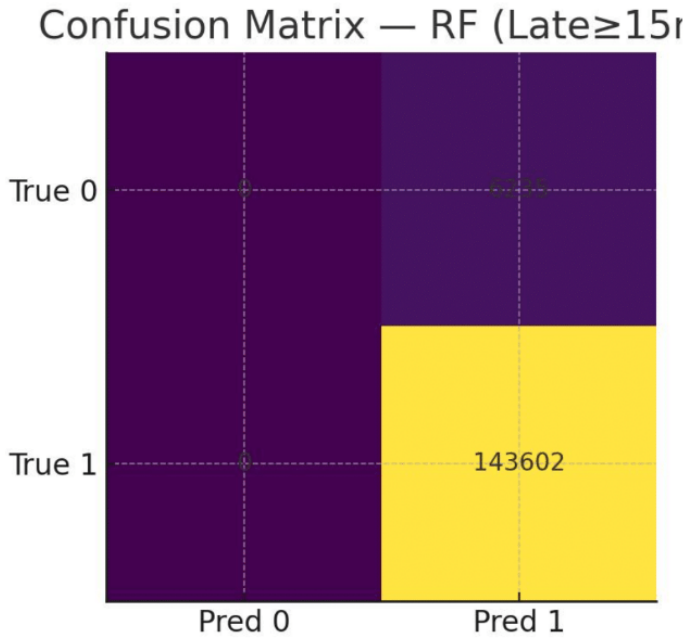

Figure 10: Confusion Matrix — RF (Late $\geq$ 15 min)

axis represents the true labels). The color intensity encodes the number of instances, with annotated values showing the absolute counts of flights in each category: true negatives (top-left), false positives (top-right), false negatives (bottom-left), and true positives (bottom-right).

The confusion matrix indicates that the RF model correctly identifies a large number of late flights (true positives), while also maintaining a high volume of correctly predicted on-time flights (true negatives). The dominance of the diagonal cells suggests strong classification ability, with relatively fewer misclassifications. However, the presence of false positives and false negatives shows that some on-time flights are incorrectly flagged as late, and some late flights go undetected. Overall, the matrix supports the view that RF is effective for operational alerting of late arrivals.

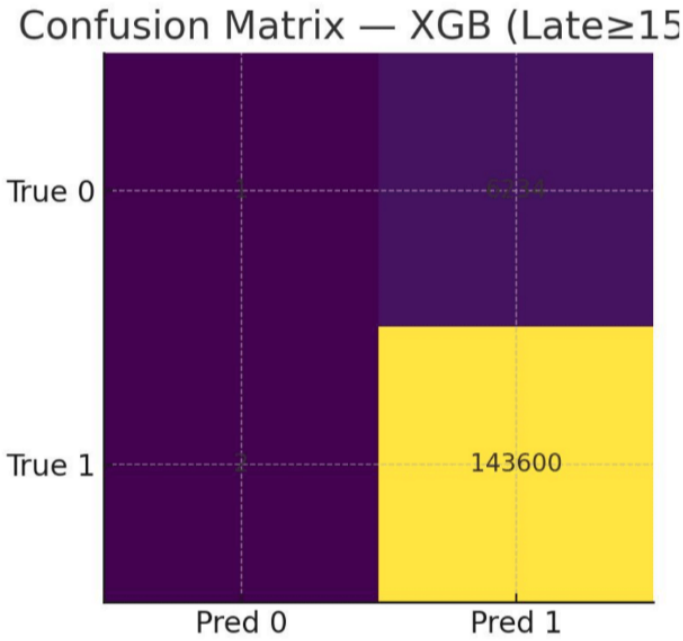

Figure 11: Confusion Matrix — XGB (Late $\geq$ 15 min)

The Figure 11 presents the confusion matrix of the XGBoost (XGB) model for the classification task of predicting whether a flight arrival is late by 15 minutes or more, The confusion matrix shows that XGB identifies a large number of late flights correctly (true positives), similar to RF, while also accurately classifying many on-time flights (true negatives). However, as with RF, there are still false positives (on-time flights flagged as late) and false negatives (late flights missed). Compared to RF, XGB displays a very close performance profile, indicating strong classification ability in this binary task. This supports the conclusion that XGB is a highly competitive alternative to RF, though RF retains a slight edge in robustness across metrics and time windows, while CAT lags further behind.

6. Conclusion

The results obtained from the different tests reveal clear and consistent trends in the comparison between Random Forest (RF), XGBoost (XGB), and CatBoost (CAT). Overall, Random Forest stands out as the best-performing and most robust model. It achieves the lowest MAE and RMSE values, as well as the highest average R^2 scores, confirming its ability to provide accurate and stable predictions. Its superiority is supported by global analyses, yearly results, and significance tests (Wilcoxon), which demonstrate that these advantages are not due to chance.

The comparative evaluation of the models across eleven significant figures shows clearly that Random Forest is the most accurate and robust model, with the lowest errors, and the highest explanatory ability, and also with superior consistency across time and constraints.

Future research should focus on several directions. First, extending the analysis to hybrid or ensemble methods combining RF and XGB could reveal additional gains in predictive accuracy and robustness. Second, calibration techniques and threshold optimization should be explored to enhance classification stability in operational settings. Third, applying deep learning models such as recurrent or attention-based architectures may provide new insights into temporal dependencies in flight delays. Finally, testing the models on larger and more heterogeneous datasets, including international flights and real-time weather data, would ensure broader generalizability and operational readiness.

Overall, RF remains the most reliable choice today, XGB is a pragmatic alternative with computational efficiency, and CAT appears less suitable in this specific context. Nonetheless, future methodological and data-oriented advances could reshape this hierarchy.

Declaration on Generative AI

During the preparation of this paper, the first author used ChatGPT-5 in order to verify information and enhance the text. After using this tool, the edition was done by the first author as needed, and he takes full responsibility for the final content.

References

[1] J. J. Rebollo, H. Balakrishnan, Characterization and prediction of air traffic delays, Transportation research part C: Emerging technologies 44 (2014) 231–241.
[2] European Organisation for the Safety of Air Navigation (EUROCONTROL), Performance review report: An assessment of air traffic management in europe, 2019.
[3] K. Gopalakrishnan, H. Balakrishnan, A comparative analysis of models for predicting delays in air traffic networks, 2017.
[4] S. Choi, H. Kim, J. Lee, Prediction of flight delays using machine learning techniques, 2015.
[5] M. Yazdi, et al., Flight delay prediction based on machine learning with self-attention, 2020.
[6] H. Zhang, C. Song, J. Zhang, H. Wang, J. Guo, A multi-step airport delay prediction model based on spatial-temporal correlation and auxiliary features, IET Intelligent Transport Systems 15 (2021) 916–928.

[7] M. Mamdouh, M. Ezzat, H. A. Hefny, A novel intelligent approach for flight delay prediction, Journal of Big Data 10 (2023) 179.
[8] L. Wang, J. Mao, L. Li, X. Li, Y. Tu, Prediction of estimated time of arrival for multi-airport systems via "bubble" mechanism, Transportation Research Part C: Emerging Technologies 149 (2023) 104065.
[9] K. Kiliç, J. M. Sallan, Study of delay prediction in the us airport network, Aerospace 10 (2023) 342.
[10] I. Hatıpoğlu, O. Tosun, Predictive modeling of flight delays at an airport using machine learning methods, Applied Sciences 14 (2024) 5472.
[11] J. Ajayi, Y. Xu, L. Li, K. Wang, Enhancing flight delay predictions using network centrality measures, Information 15 (2024) 559.
[12] F. Cao, et al., Predicting flight arrival times with machine learning and multiple feature sets, 2024.
[13] J. Pineda-Jaramillo, C. Munoz, R. Mesa-Arango, C. Gonzalez-Calderon, A. Lange, Integrating multiple data sources for improved flight delay prediction using explainable machine learning, Research in Transportation Business & Management 56 (2024) 101161.
[14] W. Gao, D. Pang, Airport time profile construction driven by flight delay prediction, Scientific Reports 14 (2024) 18715.
[15] M. Dai, et al., A hybrid ml-based model for predicting flight delays (cowrf), 2024.
[16] M. Beltman, M. Ribeiro, J. de Wilde, J. Sun, Dynamically forecasting airline departure delay probability distributions for individual flights using supervised learning, Journal of Air Transport Management 126 (2025) 102788.
[17] N. A. Ribeiro, J. Tay, W. Ng, S. Birolini, Delay predictive analytics for airport capacity management, Transportation Research Part C: Emerging Technologies 171 (2025) 104947.
[18] Y. Yuan, Y. Wang, C. S. Lai, Multi-attribute data-driven flight departure delay prediction for airport system using deep learning method, Aerospace 12 (2025) 246.
[19] A. Cook, G. Tanner, Flight delay and aviation performance: A comprehensive review, 2021.
[20] X. Xu, et al., A survey on machine learning for air traffic management, 2022.