Forecasting Daily Departing Passenger Throughput at JFK Airport

Team Flight 110: Hao Zhu, Diana Aleksieieva, Hong Zhao
DSE I2100 Applied Machine Learning and Data Mining

Spring 2026 Final Report — May 20, 2026

Abstract

JFK International Airport screens about 60,000 to 110,000 departing passengers per day, so accurate short-term forecasting can support TSA and Port Authority staffing decisions. We model two-day-ahead passenger throughput as a supervised regression problem using TSA checkpoint counts [1], NOAA weather records [2, 3], BTS scheduled departures [4], and calendar features [6, 7]. Models are evaluated on a fixed 2025 holdout set. We compare a seasonal-naive baseline with nine learned models: SARIMA/SARIMAX, Ridge, SVR, Random Forest, XGBoost [8], LightGBM [9], LSTM [10], and TabPFN [11]. The best fixed-holdout model is TabPFN with feature set D, which uses 82 predictors and achieves 3.66% MAPE and 0.835 R^2. A daily retrained walk-forward TabPFN reaches 3.31% MAPE and is reported as an operational upper bound. Among other models, LightGBM performs best with 4.46% MAPE. Ablation results show that autoregressive lag features contribute the most, reducing MAPE by about 2.0 percentage points, followed by calendar features at about 0.7 points. Weather, COVID, and flight-operation features add limited marginal value. Error analysis identifies rare disruptions, holiday-window timing errors, weather effects without recorded cancellations, and weekday-specific demand mismatches as the main failure modes.

1 Introduction

JFK International Airport is one of the busiest international gateways in North America, with over 63 million passengers in 2024. TSA and Port Authority teams need to plan staffing before passengers arrive, so accurate short-term passenger forecasts can help reduce long security lines, missed flights, and inefficient staffing.

This forecasting task is difficult because passenger demand is not fully regular. Volume changes by weekday, season, holidays, and school breaks. It can also shift suddenly because of weather, cancellations, and other airport disruptions. Simple seasonal rules, such as using the same day last week, cannot capture all of these interactions. We frame daily JFK TSA checkpoint throughput as a supervised regression problem. Given information available on day t, the goal is to predict total

departing TSA checkpoint volume two days ahead, on day $t + 2$. This horizon reflects a practical staffing window for airport operations. The model uses recent passenger history, calendar variables, weather records, weather-warning signals, and scheduled flight activity.

Machine learning is appropriate because demand patterns are nonlinear and context-dependent. For example, a Friday before Christmas can behave very differently from a regular Friday in February, and a storm day can break normal seasonal patterns. In our exploratory analysis, day-of-week alone explains less than 60% of the variance, showing that additional temporal, calendar, and disruption-related features are needed. This project is organized around five research questions:

RQ1. Which model family gives the best two-day-ahead forecast on the 2025 holdout set?

RQ2. Which feature groups add the most value, and which are operationally redundant?

RQ3. What causes the largest forecast errors, and can disruption signals such as weather warnings or cancellations explain them?

RQ4. How far ahead can we forecast before accuracy degrades meaningfully?

RQ5. How can the best model be presented to stakeholders for real operational use?

To answer these questions, we build a leakage-controlled daily forecasting pipeline, compare statistical, classical machine-learning, deep-learning, and tabular foundation models, and analyze both aggregate accuracy and operational failure modes. We also evaluate feature-group ablations, forecast horizons beyond $t + 2$, and a walk-forward setting that approximates daily operational retraining.

2 Related Work

Airport passenger forecasting connects transportation planning, airport operations, and time-series machine learning. Prior work shows that neural and ensemble models can outperform classical ARIMA-family baselines during unstable travel periods [12], and that handcrafted calendar and lag features are important for airport passenger-throughput prediction [13].

Our target signal is motivated by BTS work that estimates airport enplanements from TSA checkpoint inspections using correlation analysis and linear regression [14]. We also use PANYNJ traffic statistics as an external sanity check against broader JFK passenger-volume trends [15]. Finally, we evaluate TabPFN as a modern tabular foundation-model baseline for this forecasting task [16].

This project extends prior work by building a unified daily dataset from TSA, NOAA, BTS, IEM, and calendar sources [1–7]; evaluating all models on the same fixed 2025 holdout period; running structured feature-group ablations; and comparing statistical models, traditional machine learning, deep learning, and a tabular foundation model on the same two-day-ahead JFK forecasting task.

3 Data

3.1 Sources and Licensing

Table 1 summarizes the public data sources used to build the daily JFK forecasting panel. All sources are publicly downloadable, and no IRB approval, NDA, or paid access was required. Provenance notes and reload instructions are documented in `data/README.md`.

Table 1: Data sources, temporal coverage, licensing, and record counts used in the JFK passenger-throughput forecasting dataset.

Source	Cadence	Period	License	Records
TSA Throughput (mikelor)	Hourly	Dec. 2018–May 2025	MIT	56,080 hourly
PANYNJ Airport Traffic Data	Monthly	Jan. 2000–Jan. 2026	Public domain	2,284,947
NOAA GHCND (JFK, USW00094789)	Daily	Jan. 2017–Feb. 2026	Public domain	3,319 daily
BTS TranStats (JFK departures)	Daily	Jan. 2019–Jun. 2025	Public domain	2,373 daily

3.2 Unified Daily Panel

All sources are aligned to a single daily index. The final modeling panel contains about 1,980 daily observations and up to 82 engineered features. The prediction target is `daily_throughput`, defined as the total daily TSA passenger count across five JFK terminal groups. Terminals 2 and 4 are merged to handle the 2023 Terminal 2 closure consistently. The forecast horizon is fixed at $t+2$. To avoid target leakage, lag features start at lag 2, and rolling statistics use `shift(2)`. This ensures that features for day t use only information available by day t.

Calendar features include cyclical encodings for day of week, month, and day of year, plus raw integer calendar variables for tree-based models. We also add holiday-distance counters, long-weekend flags, and multi-day indicators for Thanksgiving, Christmas, July 4, and spring break. To capture yearly seasonality, we use a smoothed same-weekday-last-year feature instead of a single `lag_365`. This reduces sensitivity to one-day anomalies, but removes about 365 warmup rows.

Missing values are limited. Eight missing TSA days from July 2022 and November 2024 are imputed with `shift(7)`. About 1% of missing weather values are forward-filled.

COVID-era demand changes are represented with two binary indicators: `covid_acute` for March–June 2020 and `covid_recovery` for July 2020–May 2022. SARIMAX is trained only on the post-recovery period starting May 30, 2022. Machine-learning models use the full 2019–2024 training period with COVID indicators to model the regime shift.

3.3 Feature Sets

Table 2 shows the nested feature sets used to match each model family with an appropriate input representation.

Table 2: Feature-set design for model comparison, including column counts, assigned model families, and added feature groups.

Set	Cols	Used by	Notes
A	34	SARIMAX	Exogenous features only; SARIMAX has built-in AR/MA terms.
B	69	Ridge, SVR, LSTM	Set A plus lag, rolling, difference, and smoothed same-weekday-last-year features.
C	72	Random Forest, XGBoost, LightGBM	Set B replace sin/cos calendar variables to raw integer ones and add five scheduled-departure difference features.
D	82	TabPFN	Set B plus six additional calendar features and three holiday-distance counters.

3.4 Splits and Cross-Validation

We use a fixed calendar-date split. The training set includes all observations from January 1, 2020 to December 31, 2024, and the test set is the full 2025 calendar year. This creates a 365-day holdout period that is fully separated from model selection.

Hyperparameter tuning uses three year-aligned time-series cross-validation folds. In each fold, earlier years are used for training and the next year is used for validation. The 2025 test year is never used during tuning. All preprocessing steps, including standardization and imputation, are fit only on the training fold and then applied to the validation fold.

After tuning, each model is refit once on the full training period and evaluated on the 2025 holdout. This setup prevents test-set leakage and makes results comparable across model families. It also tests generalization under distribution shift because the training period includes the COVID collapse and recovery, while 2025 is fully post-recovery.

3.5 EDA Headlines

Exploratory analysis shows that JFK TSA throughput has strong temporal structure. Daily volume ranges from 1,640 passengers during the April 2020 COVID trough to 114,397 passengers during the summer 2024 peak. The mean is about 70,000 passengers per day, with large variation by weekday, month, and recovery period. Recent demand is the strongest predictor. Figure 1 shows a very strong relationship between daily throughput and the 7-day rolling mean. This supports the use of lag and rolling-window features in the supervised models.

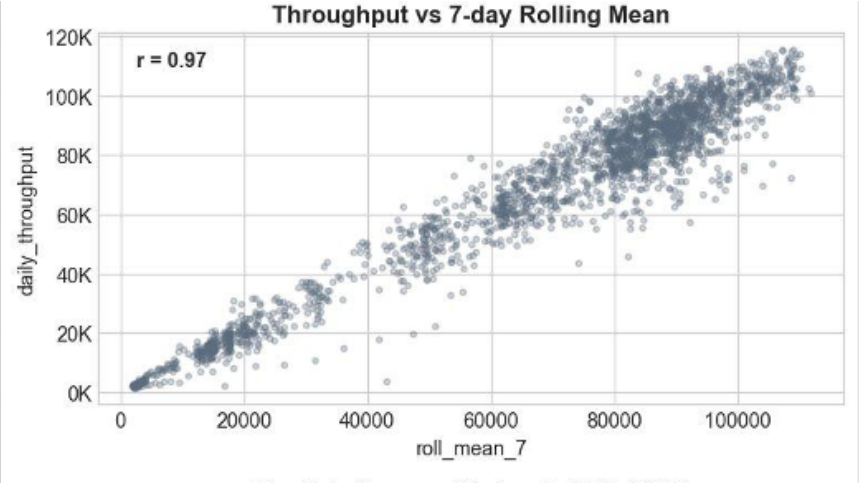


Figure 1: Daily TSA throughput versus the 7-day rolling mean. The strong correlation shows that recent passenger volume is highly informative for short-term forecasting.

Yearly seasonality is also important. Figure 2 shows that same-weekday-last-year throughput is strongly related to current throughput in the post-recovery period. Because a single one-year lag can be sensitive to abnormal days, we use a smoothed same-weekday-last-year feature.

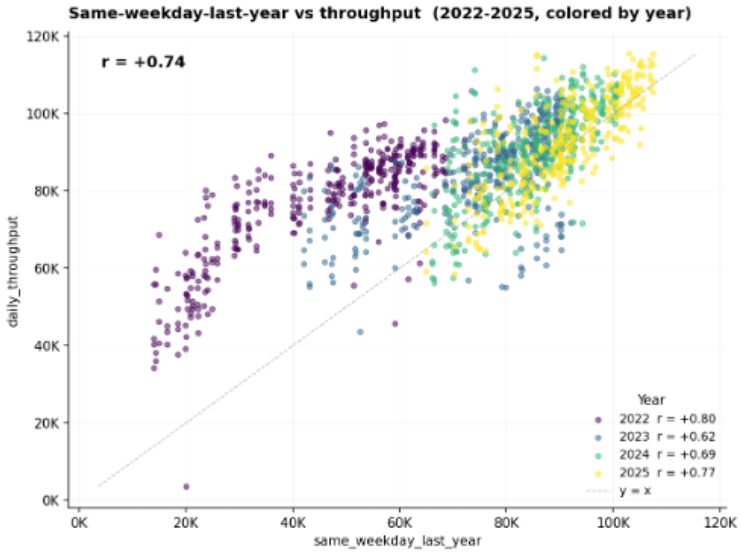


Figure 2: Same-weekday-last-year throughput versus current daily throughput, colored by year. The relationship supports using yearly seasonal features, while the year-level shifts show the effect of post-COVID recovery.

Calendar patterns are visible at weekly and monthly scales. Figure 3 shows that Thursday and Sunday are usually the busiest days, while Tuesday and Wednesday tend to be lower. Monthly seasonality is also clear, with July and August as peak travel months and January and February as lower-demand months.

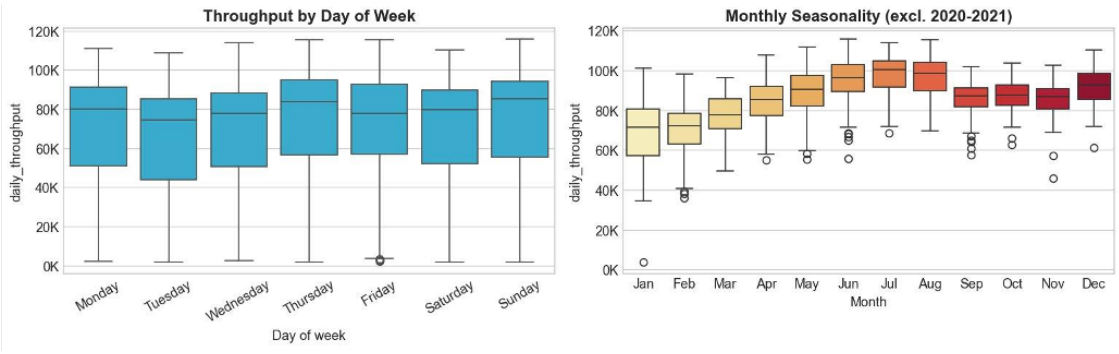


Figure 3: Daily TSA throughput by day of week and month. The distributions show clear weekly and annual seasonality in passenger demand.

4 Methods

We compare ten forecasting approaches across three model families: statistical time-series models, classical machine-learning models, and deep/foundation models. All models predict two-day-ahead JFK TSA throughput and are evaluated on the same fixed 2025 holdout set when possible.

4.1 Baselines and Statistical Models

We use a seasonal-naive lag-7 ensemble as the main baseline. It predicts future throughput from recent weekly patterns and provides a simple reference for measuring the value of learned models.

We also evaluate SARIMA and SARIMAX as classical time-series models. SARIMA captures autoregressive, moving-average, and weekly seasonal structure. SARIMAX extends this framework with exogenous predictors, including calendar variables, holidays, weather, COVID indicators, and scheduled departures. The final SARIMAX model is trained on the post-recovery period beginning May 30, 2022 and evaluated in a rolling two-day-ahead setup.

4.2 Classical Machine-Learning Models

We evaluate Ridge regression, SVR, Random Forest, XGBoost [8], and LightGBM [9] on the engineered feature sets. Hyperparameters are selected using the training period and year-aligned validation folds, with the 2025 holdout reserved for final testing.

Ridge and SVR test how much of the signal can be captured with relatively interpretable models. Random Forest provides a bagged-tree baseline. XGBoost and LightGBM represent boosted-tree approaches, which are expected to capture nonlinear interactions between recent passenger history, calendar effects, weather, and flight-schedule features. LightGBM is evaluated across multiple random seeds to check stability.

4.3 Deep and Foundation Models

We evaluate two higher-capacity models: an LSTM [10] sequence model and TabPFN [11, 16]. The LSTM uses a 14-day history window with feature set B and is trained with early stopping. Final inference uses a five-seed ensemble.

TabPFN is evaluated on feature set D, the broadest feature representation with 82 predictors. Unlike gradient-trained models, TabPFN performs in-context tabular prediction by attending over training examples during inference. We tune a small grid over internal estimators, softmax temperature, averaging strategy, and optional feature selection. The final production prediction averages three seeds. Uncertainty is estimated with a moving-block bootstrap using block size 14 and 2,000 bootstrap samples.

We also evaluate a walk-forward TabPFN setting. For each 2025 target date, the model is refit using only data available two days before the prediction date. Because this setting uses repeated daily refitting, we report it separately as an operational upper bound rather than as the main fixed-split comparison.

4.4 Stacking Ensemble

We test whether combining the strongest model families improves performance. The stacking ensemble uses SARIMAX, LightGBM, and TabPFN as base models, with Ridge regression as the meta-learner. We compare base predictions alone, base predictions plus Set-A exogenous features, and base predictions plus broader autoregressive features on the October–December 2025 meta-test window.

The stacking model is mainly used to study ensemble behavior and uncertainty calibration. Its results are reported separately because the meta-test period is shorter than the full 2025 holdout.

4.5 Evaluation Protocol

We evaluate all models using MAE, RMSE, MAPE, and R^2. MAPE is the primary metric because it is easy to interpret for airport staffing decisions. MAE and RMSE report error in passenger counts, while R^2 measures explained variance. The 2025 holdout set is used only for final evaluation. Hyperparameters are selected using year-aligned validation folds, and preprocessing is fit only on training folds to avoid leakage. For stochastic models, we report per-seed variation. Where appropriate, we also report bootstrap confidence intervals.

5 Experiments and Results

The full experiment log is stored in `experiments/README.md`. It records final model configurations, selected hyperparameters, cross-validation MAPE, 2025 holdout metrics, and notebook ownership. Per-seed metric files and Optuna [18] trial logs provide supporting evidence for stochastic models and tuning runs.

5.1 Headline Comparison

Table 3 compares all final models on the 2025 holdout set. Lower MAE, RMSE, and MAPE are better; higher R^2 is better.

Table 3: Headline model comparison on the 2025 holdout set. Brackets show 95% confidence intervals for MAPE where available.

Rank	Model	MAE	RMSE	MAPE (%)	R^2	Test n
1	TabPFN, Set D, static 3-seed	3,359	4,875	3.66 [3.10–4.11]	0.835	365
2	LightGBM, MAE objective	4,050	5,344	4.46 [4.06–4.88]	0.801	365
3	SARIMAX, $h = 2$ rolling	4,126	5,605	4.64 [4.18–5.14]	0.782	364
4	XGBoost	4,507	5,904	4.90 [4.46–5.35]	0.758	365
5	LSTM, 5-seed ensemble	4,973	6,393	5.41 [4.95–5.88]	0.716	365
6	Ridge regression	5,240	6,725	5.73 [5.26–6.24]	0.685	365
7	Random Forest	5,798	7,219	6.34 [5.86–6.83]	0.638	365
8	SVR	5,616	7,311	6.40 [5.80–7.02]	0.628	365
9	Naive, lag-7 ensemble	6,642	8,814	7.42	0.460	365
–	TabPFN walk-forward $t + 2$	2,996	4,520	3.31 [2.74–3.78]	0.858	365
†	Stacking, Ridge Mode 2	3,950	5,281	4.10	0.633	92
–	MSTL + SARIMAX	5,084	6,996	5.72	0.660	365

† The stacking model is evaluated only on October–December 2025 because the meta-learner is trained on January–September 2025 base-model predictions.

TabPFN is the best model on the fixed 2025 holdout, reducing MAPE from 6.47% for the seasonal-naive baseline to 3.66%. This is a 51% relative reduction in percentage error. LightGBM is the strongest traditional machine-learning model, followed closely by SARIMAX. The boosted-tree and SARIMAX results show that much of the forecasting signal can be captured with engineered autoregressive and calendar features, while TabPFN benefits most from the full feature set. The walk-forward TabPFN result is reported separately because it refits the model for each target date using the most recent available training pool. It represents an operational upper bound rather than a direct fixed-model comparison. The stacking result is also not directly comparable because it is evaluated only on the October–December 2025 meta-test window. † The stacking model is evaluated on October–December 2025 only (n=92) because the meta-learner is trained on January-September 2025 base-model predictions. It should be interpreted as an ensemble deployment result, not as a full-year headline comparison. Figure 5 visualizes the same headline comparison as Table

3. The error bars show the bootstrap 95% confidence interval for full-year MAPE. TabPFN has the lowest error and the strongest R^2, followed by LightGBM and SARIMAX.

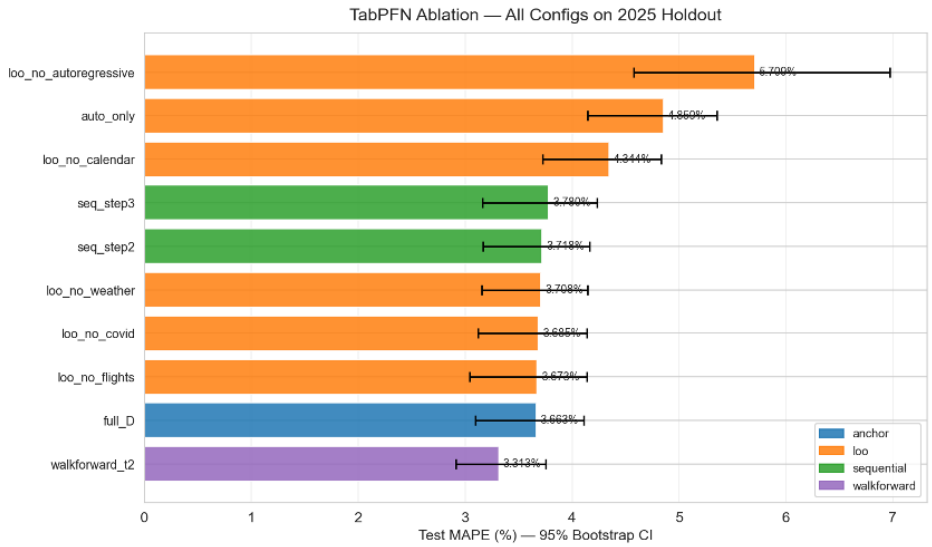


Figure 4: Full-year 2025 model comparison by MAPE with bootstrap 95% confidence intervals. Models are grouped by family, and labels report both MAPE and R$ˆ2$.

Headlines. All learned models improve over the seasonal-naive lag-7 ensemble baseline. The best fixed-holdout model is TabPFN, which reduces MAPE from 7.42% to 3.66%. This is a 3.76 percentage-point absolute improvement and about a 51% relative reduction in error. Figure 6 shows the full-year prediction trace for the best fixed-holdout model, TabPFN. The model closely follows the main 2025 demand pattern, including the winter trough, summer peak, and late-year increase. This visual result is consistent with the headline metrics: TabPFN achieves the lowest full-year MAPE, 3.66%, and the highest R^2=0.835. Most observed values fall within the prediction band, showing that the model captures regular calendar and autoregressive structure well. The remaining large misses occur mainly around abrupt demand changes, such as disruption days and rapid post-holiday shifts, which are harder to predict from historical and calendar features alone.

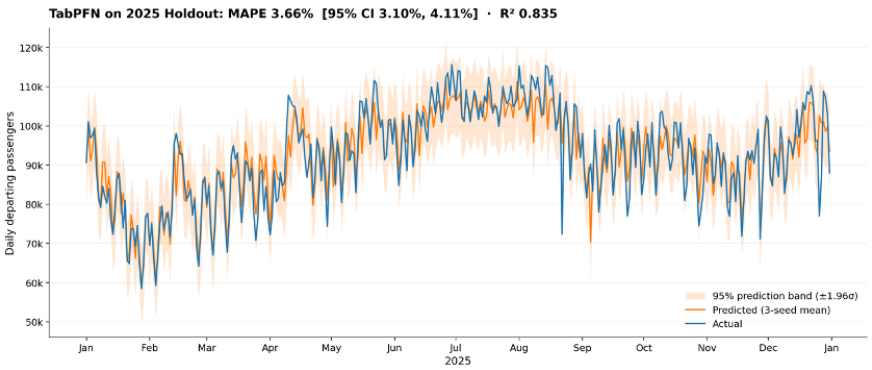


Figure 5: TabPFN Set D predictions on the 2025 holdout. The plot compares actual daily TSA throughput with the 3-seed mean prediction and a 95% prediction band.

5.2 Per-Seed Stability

For stochastic models, we measure sensitivity to random initialization by repeating inference across multiple seeds. TabPFN is evaluated with 3 seeds, LSTM with 5 seeds, and LightGBM with 10 seeds.

Table 4: Per-seed stability for stochastic models.

Model	Mean MAPE	Per-seed std.
TabPFN, Set D	3.66%	0.06 pp
TabPFN walk-forward	3.31%	0.008 pp
LightGBM, MAE objective	4.55%	0.05 pp
LSTM	5.41%	0.05 pp

Seed sensitivity is small compared with the gaps between model families. For example, the difference between TabPFN and LightGBM is about 0.8 percentage points, while their per-seed standard deviations are only about 0.05–0.06 points. This suggests that the headline ranking is not driven by random seed noise.

5.3 Error Patterns by Weekday and Month

We also examine whether errors concentrate on specific weekdays or months. TabPFN improves over the seasonal-naive baseline on every weekday, but Friday remains the hardest case. Friday MAPE is 6.76%, compared with 2.66–3.38% for most other weekdays. The walk-forward TabPFN reduces Friday error to 5.43%, but does not remove the pattern completely. Monthly errors show a similar concentration. July, November, and February are the easiest months, with MAPE below 3%. April and August are the hardest, with MAPEs of 4.88% and 4.56%, respectively. April likely reflects spring break and pre-Easter timing effects, while August includes disruption effects such as Hurricane Erin. These results suggest that the model performs best during regular seasonal periods

and struggles more when demand is affected by irregular calendar timing or operational shocks.

5.4 Feature Importance Across Models

We compare feature-importance signals across model families using the most appropriate method for each model: tree-based importance for Random Forest, XGBoost, and LightGBM; standardized coefficients for Ridge and SARIMAX; permutation importance for SVR, LSTM, and TabPFN. Across models, the most consistently important predictors are autoregressive and seasonal features. The 7-day rolling mean, 7-day rolling maximum, terminal-level lag features, and same-weekday-last-year features appear near the top for most model families. SARIMAX places the highest weight on the smoothed same-weekday-last-year feature, followed by scheduled departures. Ridge also relies heavily on terminal-level lags, which suggests that once recent-history features are exposed directly, linear models can capture much of the predictable structure. Different model families use these signals in different ways. Tree-based models tend to rank rolling minimum and maximum features highly because they create useful split points. Linear models rely more directly on raw lag values. SVR gives high importance to 7-day difference features, while these features are less dominant in other models. As shown in Figure 7, the top-ranked features differ by model family, but the same broad pattern appears across models: autoregressive features, rolling statistics, and calendar variables dominate the feature-importance rankings.

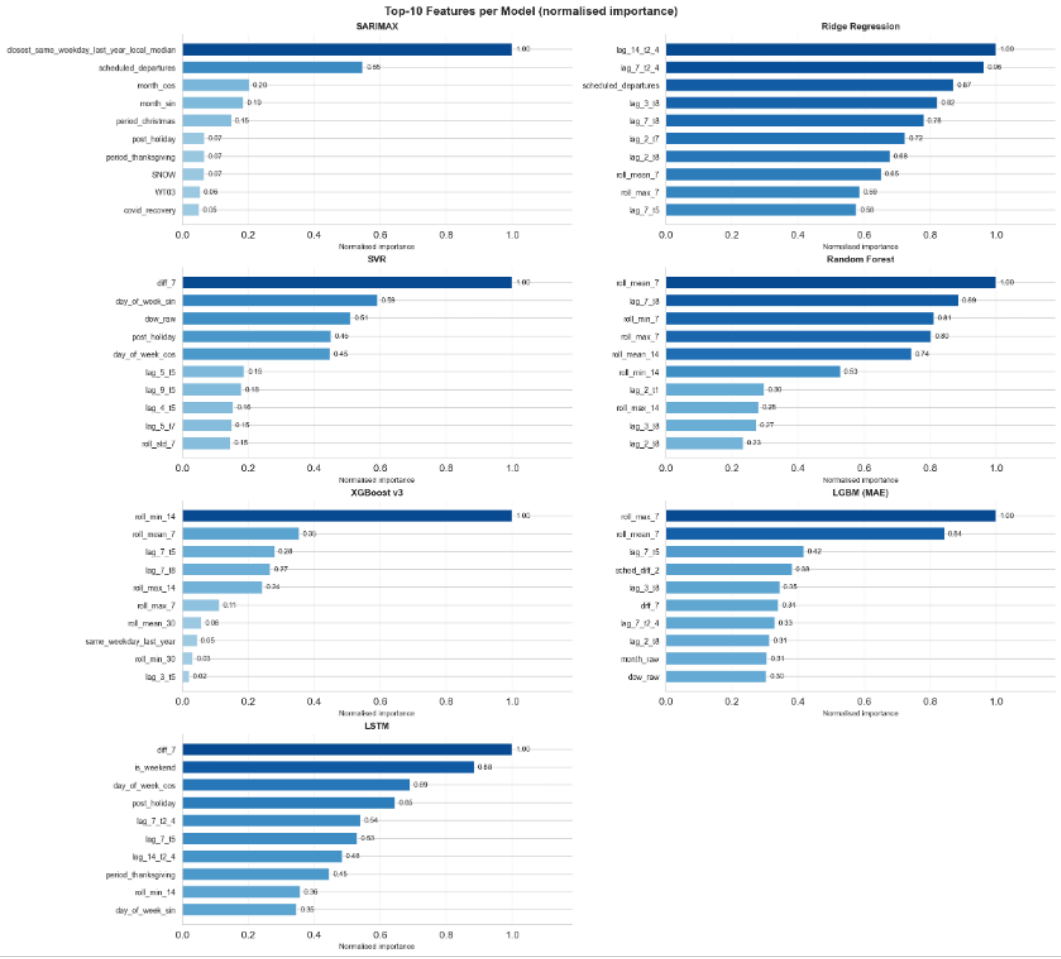


Figure 6: Top-10 features by aggregate cross-model importance, with per-model contributions stacked. Autoregressive and calendar features dominate.

5.5 Residual Diagnostics

Because this is a regression task, we inspect residuals for the best fixed-holdout model, TabPFN Set D. Figure 8 shows three diagnostics for the 2025 holdout: residuals over time, residuals versus predicted values, and the residual distribution. The residuals are approximately centered around zero, with a mean of about -180-180-180 passengers per day. This is small relative to the model MAE, suggesting no strong systematic bias. The residual-versus-predicted plot shows mild heteroskedasticity at high predicted volumes. In other words, summer peak days tend to have larger absolute errors than lower-volume winter days. This supports our use of MAPE as the primary metric, because relative error is more stable across demand levels. The largest negative residuals are concentrated around known disruption periods, including the Hurricane Erin week in August 2025 and the late-December winter-storm window. The residual histogram is roughly symmetric but has a heavier left tail, meaning the largest errors are more often overpredictions. This pattern is consistent with operational disruptions: when weather or cancellations reduce actual demand,

the model may still predict near-normal passenger volume.

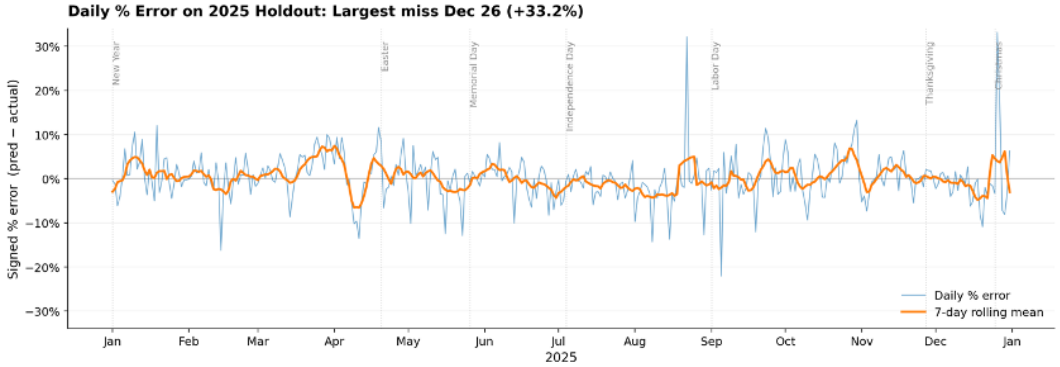


Figure 7: TabPFN Set D residual diagnostics on the 2025 holdout. Left: residuals over time, with large-error days marked. Center: residuals versus predicted values. Right: residual distribution.

We also inspect SARIMAX residuals as a comparison. SARIMAX shows the same broad failure pattern but with wider spread, especially on high-throughput summer days. Both TabPFN and SARIMAX show larger residuals on Fridays, matching the weekday error analysis in Section 5.3. This suggests that Friday demand may need a separate recalibration layer in future work.

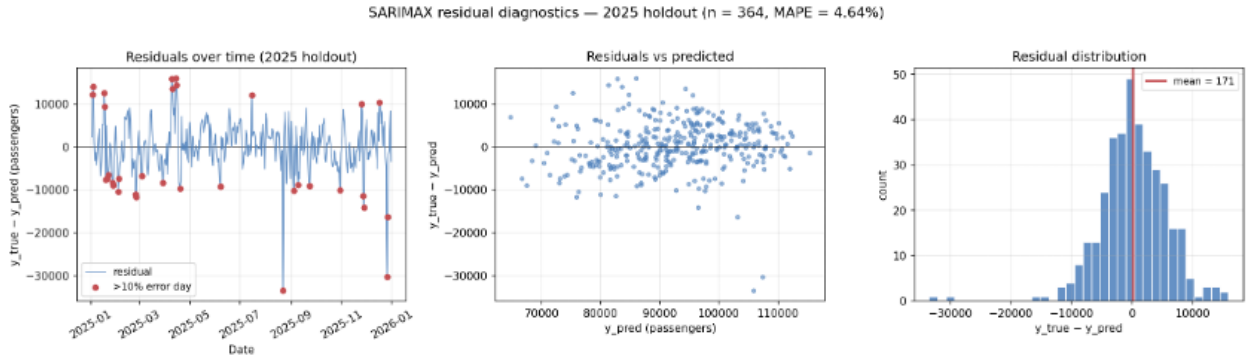


Figure 8: SARIMAX residual diagnostics on the 2025 holdout. The residual spread is wider than TabPFN, especially at high predicted volumes, and disruption days contribute to the left tail.

5.6 Hyperparameter Sensitivity

We tune the gradient-boosting models with Optuna [18] and evaluate model sensitivity using year-aligned cross-validation. The full trial logs are saved with the experiment artifacts. Figure 10 summarizes the LightGBM search. The LightGBM search converges quickly. The running-best validation MAPE falls sharply within the first 30 trials, from about 7.1% to below 5.0%. Later trials provide only small refinements, suggesting that the 200-trial budget is sufficient. The most influential LightGBM parameters are tree-structure controls. num_leaves and min_data_in_leaf have the strongest association with validation MAPE. max_depth and learning_rate have moderate in-

fluence. Regularization and sampling parameters, such as lambda_l1, lambda_l2, feature_fraction, and bagging_fraction, have much smaller effects. This suggests that model capacity and leaf regularization matter more than shrinkage penalties for this dataset.

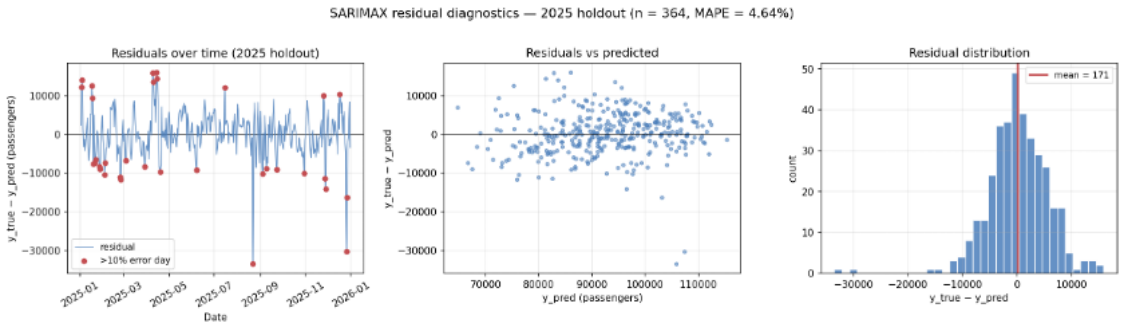


Figure 9: LightGBM hyperparameter sensitivity over 200 Optuna trials. The left panel shows validation MAPE by trial with the running-best curve. The right panel ranks hyperparameters by absolute Spearman correlation with validation MAPE. Tree-structure parameters dominate the search.

TabPFN is tuned with a smaller grid because it is used as an in-context foundation model rather than a gradient-trained one. We compare two pipelines, no feature selection (all 82 features) and per-fold LightGBM top-50 feature selection, and within each pipeline run the same 18-configuration grid over internal ensembling, softmax temperature, and averaging strategy. The winning combination is the no feature-selection pipeline with 8 internal estimators, softmax temperature 0.7, and averaging before softmax. The top configurations are close to each other, indicating that TabPFN is less sensitive to this grid than LightGBM is to tree-structure settings.

Table 5: TabPFN hyperparameter grid and selected configuration.

Hyperparameter	**Values searched**	**Selected value**
Internal estimators	{8, 16, 32}	8
Softmax temperature	{0.7, 0.9, 1.1}	0.7
Average before softmax	{`False`, `True`}	`True`
Feature selection	No feature selection; LightGBM top-50	No feature selection
Production seeds	{42, 7, 123}	Mean of all three

6 Discussion and Error Analysis

6.1 Operational Scorecard

In 2025, JFK averaged about 92,600 departing passengers per day. This makes percentage errors easy to translate into operational terms. A 5% error is roughly 4,600 passengers, while a 10% error

Table 6: Selected configurations for the three strongest fixed-holdout models.

Model	Search method	Feature set	Key selected configuration
TabPFN Set D	18-config grid	D, 82 features	8 estimators, temperature 0.7, average before softmax, no feature selection, 3-seed mean
LightGBM, MAE objective	Optuna TPE, 200 trials	C, 72 features	Shallow boosted trees with tuned leaf size, depth, learning rate, and sampling parameters
SARIMAX $h = 2$	AIC grid search	A, 34 exogenous features	Order $(2, 0, 2)$, seasonal order $(1, 1, 2, 7)$, post-recovery training window

is roughly 9,300 passengers. Compared with the lag-7 naive baseline, TabPFN substantially reduces the typical daily planning error. The naive baseline has an MAE of about 6,642 passengers per day, while the static TabPFN model reduces this to 3,359 passengers per day. The walk-forward TabPFN version reduces it further to 2,996 passengers per day. The remaining large errors are concentrated on a small number of disruption and holiday-edge days, where prediction intervals should be used together with the point forecast.

Table 7: Operational interpretation of model errors on the 2025 holdout.

Model	Expected absolute error	Days/year with $> 10\%$ error
Naive lag-7 ensemble	∼6,642 passengers/day	∼38 days
LightGBM, MAE objective	∼4,050 passengers/day	∼22 days
TabPFN, static	∼3,359 passengers/day	∼14 days
TabPFN, walk-forward	∼2,996 passengers/day	∼11 days

6.2 SARIMAX Failure Modes

We analyze SARIMAX large-error days by joining residuals with weather-warning records from IEM [5] and flight-cancellation rates. A bad day is defined as absolute percentage error above 15%. After excluding the noisy pre-July 2021 period, the bad-day rate is 6.81% in training and 3.02% on the 2025 holdout.

The main failure mode is disruption-driven overprediction. During storms or cancellation spikes, actual passenger volume can fall below the model's normal-demand expectation. Cancellation rate is positively correlated with absolute percentage error on bad days, confirming that operational disruptions explain part of the largest misses.

Other errors come from calendar timing and weekday-specific demand. Some large residuals occur around Easter, Passover, spring break, and post-break normalization even when cancellations are low. Friday errors also remain wider across many weeks, suggesting that the current features do not fully capture weekend-departure and school-calendar effects.

Weather warnings can also matter even when cancellations are not recorded. Therefore, zero cancellations should not be treated as "no disruption." In operation, cancellation rates, weather warnings, named storms, and calendar context should all be treated as risk signals.

6.3 TabPFN Failure Modes Across Horizons

We also analyze TabPFN errors across forecast horizons $h = 2..30$. This shows whether failures are caused mainly by longer lead time or by structurally hard target dates. The number of high-error days grows gradually with the horizon. At $h = 2$, TabPFN has 22 days above 10% absolute error and 4 days above 15%. By $h = 30h = 30h = 30$, these increase to about 37 and 14 days. This pattern shows a smooth degradation, not a sudden collapse. The more important finding is that some dates fail across many horizons. These recurring failure dates concentrate around spring break and Easter, late August weather disruption, late December winter storms, and several Fridays. These are structurally hard days: shortening the horizon helps only slightly because the model lacks the features needed to explain the event.

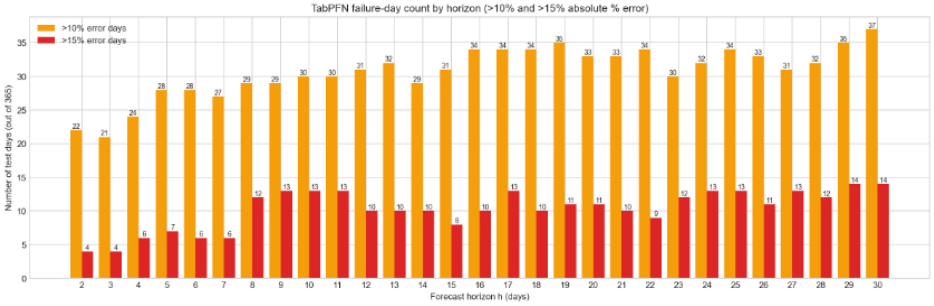


Figure 10: TabPFN failure-day counts by forecast horizon. Counts above 10% and 15% absolute percentage error increase gradually from h=2 to h=30, with no sharp cliff.

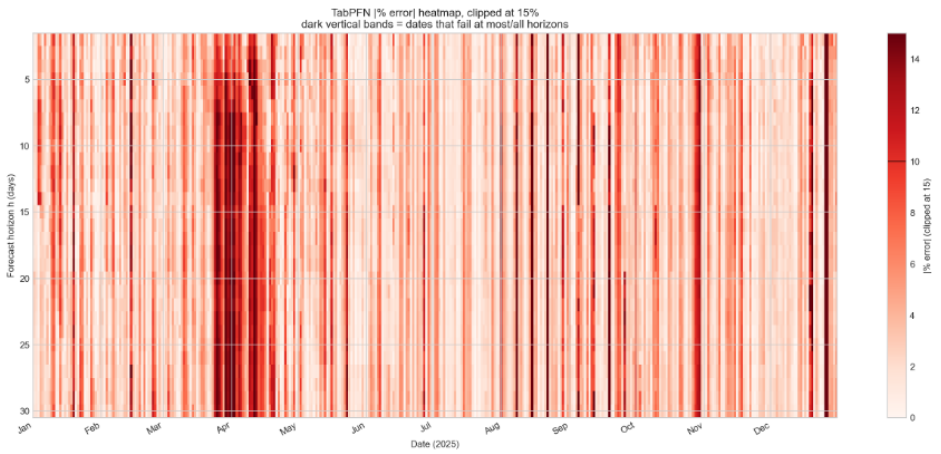


Figure 11: TabPFN absolute percentage error heatmap by date and horizon. Dark vertical bands identify dates that fail across many horizons, suggesting structurally difficult days rather than horizon-specific noise.

6.4 Why TabPFN Wins

TabPFN improves over the next-best fixed-holdout model, LightGBM, by about 0.8 percentage points in MAPE. It appears to use the same core signals more effectively in this dataset. Two observations support this interpretation. First, TabPFN performs best with the full 82-feature Set D. A LightGBM-based top-50 feature selection step did not improve performance, suggesting that TabPFN can absorb the wider feature set without overfitting as much as expected. Second, TabPFN's feature importance shows that it relies on the same broad feature groups as the other strong models: recent lags, rolling statistics, same-weekday-last-year features, and holiday-distance counters. Its improvement does not come from a completely different signal source. It comes from better use of the same autoregressive and calendar structure.

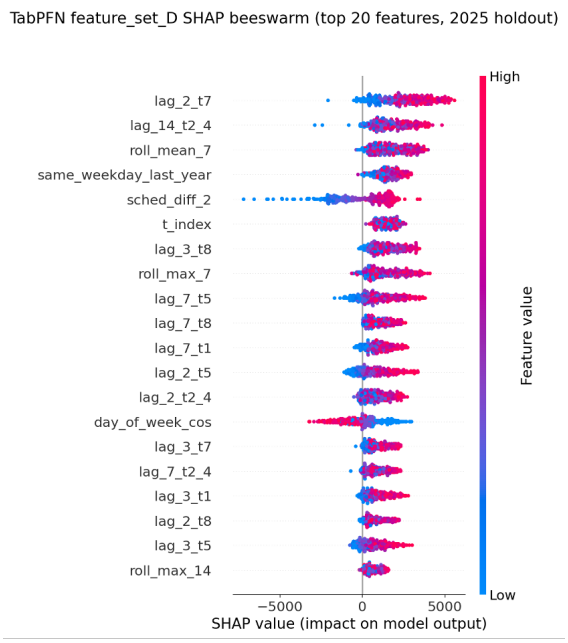


Figure 12: TabPFN SHAP [17] beeswarm for the top 20 features on the 2025 holdout. Autoregressive features dominate, followed by selected calendar and holiday-distance features. Weather and COVID features have limited direct importance.

The embedding analysis provides additional evidence for why TabPFN outperforms fixed-similarity readers. Figure 12 compares PCA projections of the raw 82-feature input and the learned TabPFN embedding. In the raw feature space, throughput values are more scattered. While TabPFN embedding space is organized with throughput values.

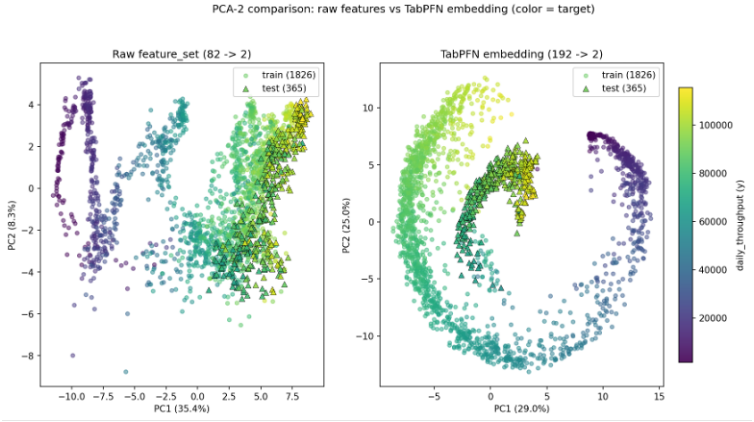


Figure 13: PCA projection of raw features versus TabPFN embedding. Points are colored by daily throughput. The TabPFN embedding shows a smoother target structure than the raw feature space, suggesting that the model learns a demand-aware representation.

Figure 13 tests whether simpler readers can reproduce TabPFN predictions from the same embedding. A fixed-k nearest-neighbor reader shows substantial spread and does not closely match

TabPFN. In contrast, a kernel-proxy reader with query-specific bandwidth aligns much more closely with TabPFN predictions. This supports the interpretation that TabPFN behaves like an adaptive similarity model: it does not just find nearby historical days, but adjusts how much local context to use for each target day.

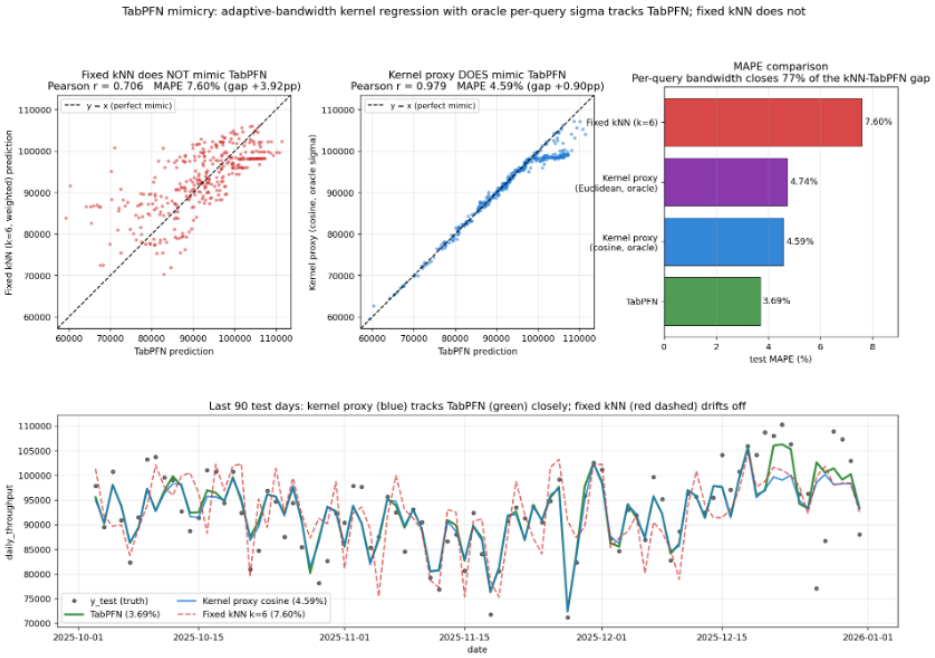


Figure 14: Reader-vs-TabPFN prediction comparison on the 2025 holdout. Fixed-kkk nearest neighbors poorly mimic TabPFN, while a query-adaptive kernel proxy closely follows TabPFN predictions. This suggests that adaptive local weighting is an important part of TabPFN's advantage.

6.5 Feature-Group Ablation

The feature-group ablation confirms that autoregressive history is the main source of predictive signal. Removing lag, rolling, and difference features increases test MAPE by about 2.05 percentage points and reduces R^2 from 0.835 to 0.701. Calendar features are the second most important group; removing them increases MAPE by about 0.68 percentage points. Weather, COVID, and flight-operation features each add less than 0.05 percentage points on the 2025 holdout when removed individually. This does not mean these signals are useless. Instead, it suggests that their current representation is too coarse or partly redundant with passenger-history features. For example, scheduled-departure features show cross-validation signal but little 2025 holdout gain, indicating possible year-to-year shift in how flight schedules relate to throughput.

The ablation results are summarized visually in Figures 14–16. Figure 14 shows that test MAPE changes very little when weather, COVID, or flight features are removed, but increases sharply when calendar or autoregressive features are removed. Figure 15 shows the same pattern in a sequential-removal setting: performance remains stable after removing the weaker groups, then

Table 8: Feature-group ablation for TabPFN Set D on the 2025 holdout.

Configuration	Features	Test MAPE	Δ vs. full model	R^2
Full Set D	82	3.663	—	0.835
Remove weather	65	3.708	+0.045	0.836
Remove COVID	80	3.685	+0.021	0.833
Remove flights	77	3.673	+0.009	0.847
Remove calendar	59	4.344	+0.681	0.811
Remove autoregressive	47	5.709	+2.045	0.701
Autoregressive only	35	4.859	+1.195	0.779

degrades after calendar features are removed. Figure 16 focuses only on the autoregressive group and shows that removing longer-history features has limited effect at first, but removing short-term weekly history causes a much larger increase in MAPE.

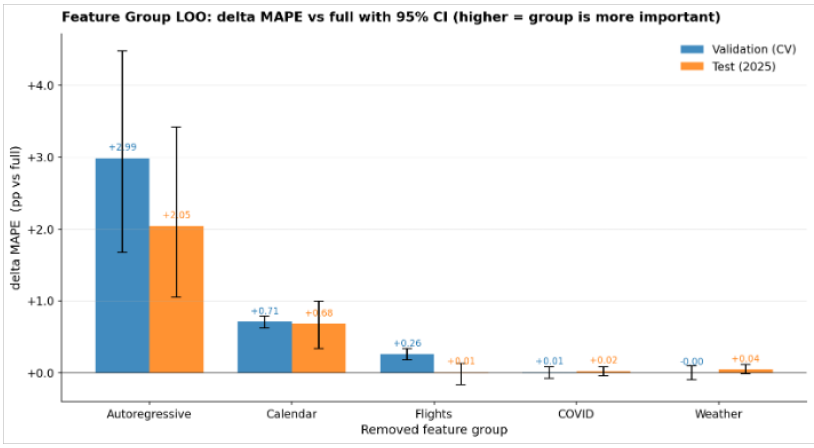


Figure 15: Leave-one-out feature-group ablation for TabPFN Set D. Bars show the increase in MAPE after removing each feature group. Autoregressive features are the most important group, followed by calendar features; weather, COVID, and flight features.

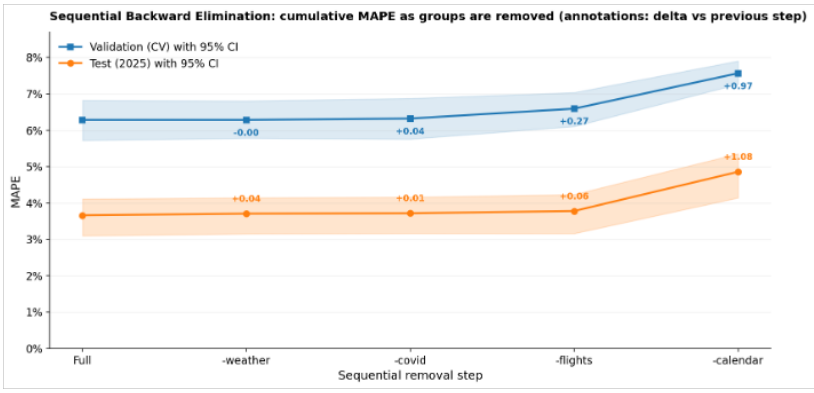


Figure 16: Sequential feature-group ablation for TabPFN Set D. MAPE remains nearly stable after removing weather, COVID, and flight features, but increases after calendar features are removed.

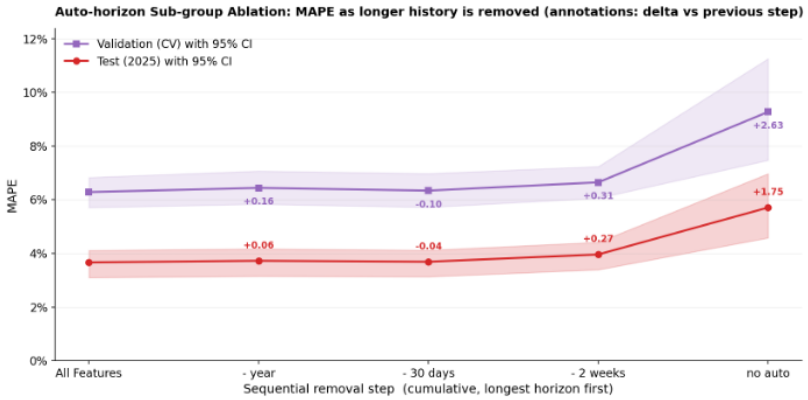


Figure 17: Autoregressive subgroup ablation. Removing longer-history features causes only small changes, while removing short-term weekly history sharply increases MAPE.

6.6 Horizon Study (3 models, h=2...{}30)

We evaluate the three strongest model families across forecast horizons from $h = 2$ to $h = 30$: TabPFN, LightGBM, and SARIMAX. The nearest-same-weekday naive baseline is included for context. Figure 17 shows MAPE and R^2 across the full horizon range. Three patterns stand out. First, TabPFN remains the best model across all horizons. It starts at 3.66% MAPE for $h = 2$ and stays below about 4.4% until the longest horizons. This suggests that TabPFN is not only useful for the two-day staffing window but also remains competitive for longer planning windows. Second, LightGBM tracks TabPFN closely, usually within about 0.3-0.6 percentage points of MAPE. Its performance is strongest at short horizons and remains relatively stable through $h = 30$. This makes LightGBM a strong classical ML alternative when interpretability, speed, or deployment simplicity is preferred. Third, SARIMAX degrades fastest as the horizon increases. Its MAPE rises from 4.64% at $h = 2$ to about 6.3% by $h = 30$. This suggests that the classical seasonal structure becomes less effective as recent autoregressive information becomes stale. The naive baseline performs the worst and deteriorates sharply after the first weekly step.

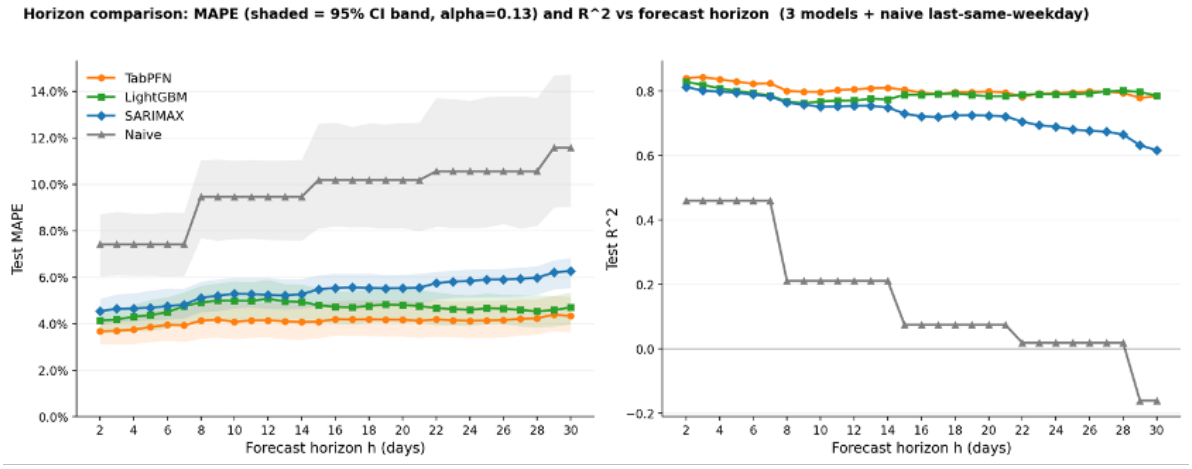


Figure 18: Forecast performance by horizon for TabPFN, LightGBM, SARIMAX, and the naive baseline. MAPE and R^2 are shown for horizons h=2 to h=30. TabPFN leads across the full horizon range, while SARIMAX degrades fastest.

Figure 18 summarizes all TabPFN ablation configurations on the 2025 holdout. The full model reaches 3.66% MAPE, while the walk-forward version improves to 3.31%. Removing weather, COVID, or flight features changes performance only slightly, but removing calendar features increases MAPE to 4.34%. Removing autoregressive features is the largest failure case, increasing MAPE to 5.71%.

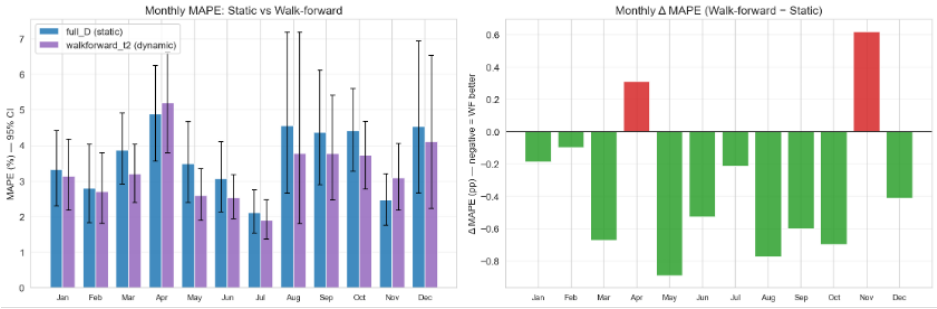


Figure 19: TabPFN ablation results across all configurations on the 2025 holdout. Walk-forward retraining improves performance, while removing autoregressive or calendar features causes the largest MAPE increase.

Figure 19 shows the same result by day of week. Friday remains the hardest day for every configuration. The walk-forward model lowers Friday error, but the gap remains visible. This suggests that Friday demand has a systematic pattern that is not fully captured by the current feature set. Removing autoregressive features increases error on every weekday, which confirms that recent demand history is important across the full week, not only on peak days.

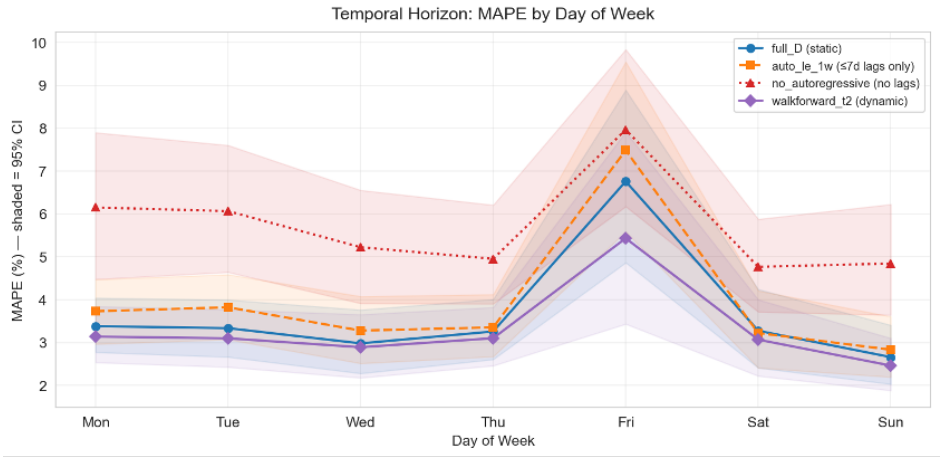


Figure 20: TabPFN MAPE by day of week for the full model, walk-forward model, auto-only model, and no-autoregressive ablation. Friday is consistently the hardest day, and removing autoregressive features increases error across all weekdays.

Figure 20 separates the importance of different autoregressive horizons. Removing long-history yearly and 30-day features has little effect. The larger jump appears when shorter weekly history is removed, and the largest degradation occurs when all autoregressive features are removed. This shows that the model depends most on recent weekly passenger patterns rather than very long historical memory.

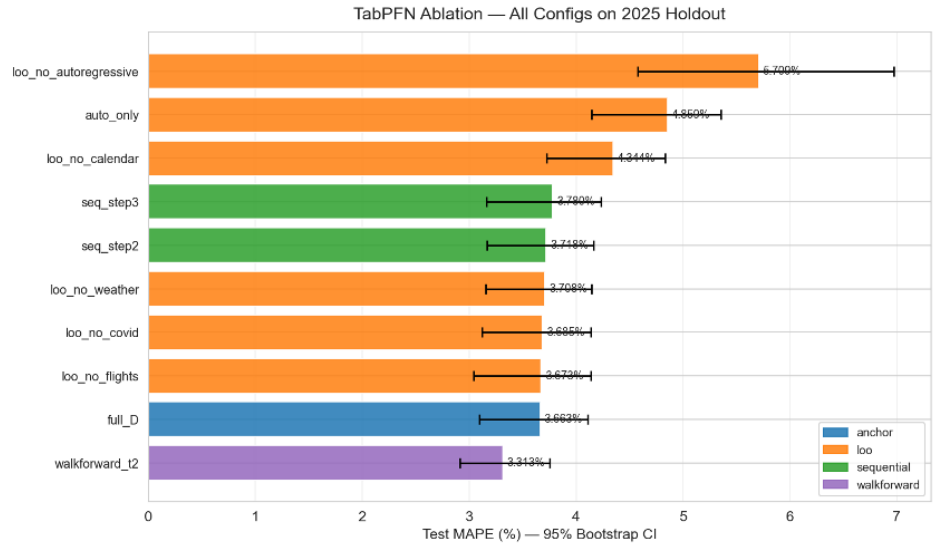


Figure 21: Autoregressive subgroup ablation. MAPE changes only slightly when longer-history features are removed, but rises sharply when short-term weekly history and all autoregressive features are removed. This indicates that recent weekly demand is the strongest predictive signal.

6.7 Walk-Forward (Operational Upper Bound)

The walk-forward $t+2$ TabPFN retrains 365 times, once for each target date. For each prediction, the training pool ends exactly two days before the target date. The static-train variant uses one production refit on the same training pool as the first walk-forward iteration, so the first walk-forward prediction matches the static prediction within seed noise.

Table 9: Static versus walk-forward TabPFN performance on the 2025 holdout.

Metric	**Static**	**Walk-forward**	**Δ**
MAE	3,359	2,996	−363
MAPE	3.66%	3.31%	−0.35 pp
R^2	0.835	0.858	+0.023
95% MAPE CI half-width	0.51 pp	0.52 pp	Similar

The walk-forward confidence interval overlaps with the static confidence interval, so the −0.35 pp improvement should be interpreted as directional rather than a hard separation. Operationally, daily retraining would be expected to lower MAPE by about 0.3 percentage points. The static-train result remains the canonical reference for cross-model comparison because the other headline models are also evaluated in a static-train setting.

6.8 Limitations and Responsible Use

This project shows strong two-day-ahead forecasting performance, but several limitations remain important for operational use. Errors are still higher on Fridays and in April, suggesting that future models need better day-of-week recalibration and richer spring-break features. Major disruptions can also cause overprediction because actual passenger volume may drop below normal demand during weather events or cancellation spikes.

The current model uses observed weather features, not real-time weather forecasts, and it predicts total JFK departing TSA throughput rather than demand by terminal, checkpoint, or shift block. Therefore, operational deployment would require forecast-time weather inputs, more granular demand modeling, and additional testing across more years and rare-event scenarios. Point forecasts should be shown with prediction intervals and reviewed by operations staff before being used for staffing decisions.

6.9 Ethics

This system is intended to support operational planning, not replace human judgment. Forecasts should not be used as fully automated staffing decisions because rare disruptions, local operations, and calendar effects may not be fully observed by the model.

A responsible deployment should focus on service quality: shorter wait times, fewer missed flights,

better shift planning, and safer passenger flow. The model should not be used as the sole justification for staffing cuts.

7 Conclusion

This project shows that daily JFK departing-passenger throughput can be forecast accurately using calendar-aware autoregressive models. We evaluated ten approaches on a fixed 2025 holdout set. The strongest fixed-holdout model is TabPFN, with 3.66% MAPE and R^2=0.835. It outperforms the best classical ML model, LightGBM at 4.46% MAPE, and the best statistical model, SARIMAX at 4.64% MAPE. The improvement is stable across random seeds and supported by bootstrap confidence intervals. The results also clarify where the predictive signal comes from. The ablation study shows that autoregressive lag features carry the largest share of information, while calendar features provide the next most important gain. Weather, COVID, and flight-operation features add limited marginal value in their current form. This does not mean they are irrelevant; rather, their current feature design is too coarse to explain rare disruption days. The main failure modes are disruption-driven overpredictions, holiday-window timing errors, weather effects without recorded cancellations, and weekday-specific demand mismatches, especially on Fridays.

References

[1] Transportation Security Administration. TSA Checkpoint Travel Numbers. Available: `https://www.tsa.gov/travel/passenger-volumes`

[2] National Oceanic and Atmospheric Administration, National Centers for Environmental Information. Climate Data Online. Available: `https://www.ncei.noaa.gov/cdo-web/`

[3] National Oceanic and Atmospheric Administration, National Centers for Environmental Information. Daily Summaries. Available: `https://www.ncei.noaa.gov/maps/daily-summaries/`

[4] Bureau of Transportation Statistics. Airline On-Time Performance Data / Airline On-Time Tables. Available: `https://www.bts.gov/explore-topics-and-geography/topics/airlines-and-airports/airline-time-tables`

[5] Iowa Environmental Mesonet. Watch, Warning, and Advisory Archives. Available: `https://mesonet.agron.iastate.edu/request/gis/watchwarn.phtml`

[6] U.S. Office of Personnel Management. Federal Holidays. Available: `https://www.opm.gov/policy-data-oversight/pay-leave/federal-holidays/`

[7] New York City Public Schools. School Year Calendar. Available: https://www.schools.nyc.gov/calendar

[8] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System." 2016. Available: https://arxiv.org/abs/1603.02754

[9] G. Ke, Q. Meng, T. Finley, et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." 2017. Available: https://github.com/microsoft/LightGBM/blob/master/docs/Features.rst

[10] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory." Neural Computation, 1997. Available: https://direct.mit.edu/neco/article/9/8/1735/6109/Long-Short-Term-Memory

[11] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second." 2022. Available: https://arxiv.org/abs/2207.01848

[12] Zachariah, A., et al. (2024). Short-term forecasting airport passenger flow during periods of volatility: Comparative investigation of time series vs. neural network models. Journal of Air Transport Management. https://www.sciencedirect.com/science/article/abs/pii/S0969699723001680

[13] Yi, S., & Guo, J. (2023). Using Machine Learning to Predict Airport Passenger Throughput. Proceedings of the 38th International Conference on Computers and Their Applications. https://easychair.org/publications/paper/bk2D/open

[14] Bureau of Transportation Statistics (2024). Preliminary Estimates for Enplanements from TSA Checkpoint Inspections Using Correlational Models and Linear Regression. https://www.bts.gov/browse-statistical-products-and-data/preliminary-estimates/preliminary-estimates-enplanements-tsa

[15] Port Authority of New York and New Jersey (2024). Air Traffic Statistics for the Port Authority Airport System. https://www.panynj.gov/airports/en/statistics-general-info.html

[16] Hollmann, N., Müller, S., Eggensperger, K., & Hutter, F. (2023). TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. ICLR 2023. https://arxiv.org/abs/2207.01848

[17] Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. NeurIPS 2017 (SHAP). https://arxiv.org/abs/1705.07874

[18] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. KDD 2019. https://arxiv.org/abs/1907.10902

A Reproducibility

All numerical results are reproducible from the project repository. Dependencies are pinned in `requirements.txt`, with optional GPU-specific packages documented separately. Random seeds are fixed for each experiment, including TabPFN seeds, the LSTM ensemble, and the LightGBM ensemble. Per-seed metrics and model outputs are saved with the report artifacts. The train/test boundary is fixed at December 31, 2024 / January 1, 2025, and cross-validation uses deterministic year-aligned folds. Preprocessing is fit only on training folds using pipeline objects where appropriate, which prevents leakage into validation or test data. The repository includes setup instructions, model-to-notebook mapping, saved predictions, feature importances, tuning logs, and run manifests.

B Experiment Tracking

The team used the assignment's Manual Markdown tables option. Every shipped model/configuration run is logged in `experiments/README.md`. Each row records the model type, key hyperparameters, training and validation metrics, test metrics, date, team member, notebook reference, and notable observations or negative results. This table is the single source of truth for the Section **??** headline comparison.

C AI Tool Usage

AI coding assistants, including Claude Code, OpenAI Codex, and GitHub Copilot, were used during development. The full disclosure and review workflow are documented in `docs/ai-usage.md`.

D Live Model Demo

We provide a public single-page demo for the TabPFN walk-forward model:

`https://jfk-forecast-demo.vercel.app/`

The demo presents persisted 2025 holdout predictions in a stakeholder-friendly format. Users can select any target date in 2025 and inspect the surrounding 7-day forecast window. The interface shows the point forecast, an empirical 90% prediction interval, and summary error metrics such as average APE, worst APE, interval coverage, and over/under prediction bias. The demo also reports $t+2$, $t+7$, $t+14$, and $t+30$ day forecasts with lower and upper 90% bounds. This connects directly to the horizon-analysis research question.