

Article

Graph-Based Multi-Horizon Forecasting of Airport Delay Propagation in the U.S. Air Transportation Network

Luís F. F. M. Santos ^{1,2,*}  and Sebastião Gonçalves ¹ 

¹ School of Management, Engineering and Aeronautics, ISEC Lisboa, Alameda das Linhas de Torres 179, 1750-142 Lisbon, Portugal; sebasgoncalves@hotmail.com

² AEROG-LAETA—Aeronautics and Astronautics Research Center, Universidade da Beira Interior, 6201-001 Covilhã, Portugal

* Correspondence: luis.santos@iseclisboa.pt

Abstract

Existing airport delay prediction models often focus on isolated flight-level or airport-level forecasts and do not explicitly represent the spatial propagation of disruptions across a national air transportation network. This study develops and evaluates a spatio-temporal graph learning framework for multi-horizon airport-level delay forecasting in the U.S. domestic network. Using 2,878,854 flight records from January 2019 to August 2023, flights are aggregated into hourly node-time tensors over 370 airports and 5334 directed connections. The proposed framework combines graph-based airport connectivity, temporal delay persistence, rare-disruption detection, and conformal uncertainty estimation over a 1–3 h forecasting horizon. The results show that LSTM attains the lowest aligned point-forecast errors, while ST-GCN provides competitive multi-horizon performance with substantially lower parameter complexity and explicit graph-aware outputs; the direct XGBoost base-lines provide nonlinear tabular benchmarks but do not dominate the final multi-horizon evaluation. These findings indicate that graph-based models are most useful not as universal point-forecast winners but as interpretable network-aware tools for monitoring delay propagation, disruption risk, and operational uncertainty.

Keywords: airport; air traffic; flight delay; neural network; artificial intelligence; graph neural network

1. Introduction

Flight delay prediction is a central operational and economic problem in the United States air transportation system. In 2024, U.S. reporting marketing carriers achieved an on-time arrival rate of 78.10%, slightly below its rate of 78.34% in 2023, meaning that approximately one fifth of scheduled operations were affected by delays or related reliability degradation [1]. The Bureau of Transportation Statistics defines a delayed flight as one arriving or departing at least 15 min later than scheduled, and reports delayed minutes only for delayed flights [2,3]. Delay costs are also substantial: FAA estimates place the annual cost of U.S. flight delays at approximately USD 33 billion when airline costs, passenger time, lost demand, and indirect costs are considered [4]. Congestion is especially relevant at major hubs such as ATL, ORD, and DFW, which were among the highest-operation U.S. airports in 2024 [5]. These facts, together with scientific evidence on systemic airport-network delay propagation, motivate short-horizon, network-aware models capable of anticipating airport-level disruption propagation rather than merely fitting isolated flight-level delays [6–8].



Academic Editors: Ján Dižo and Jose María Álvarez Rodríguez

Received: 28 May 2026

Revised: 6 July 2026

Accepted: 14 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

A delay is rarely only a local deviation from a timetable. In a highly utilized network, a late arrival can reduce turnaround buffers, delay a subsequent departure, and create downstream effects at airports that were not involved in the initial disruption. Short-term forecasts can therefore support gate allocation, recovery planning, flow management, passenger reaccommodation, and disruption triage. This operational need is consistent with the broader view that air traffic management decisions are made under uncertainty and require methods capable of representing stochastic conditions [9]. Reliability is also a key component of air transport service quality, which makes delay forecasting relevant for both operational efficiency and passenger experience [10]. Flight-level delay refers to the deviation of an individual flight from its scheduled operation. Airport-level delay is an aggregate node-time state summarizing the delay condition of an airport within a fixed hourly bin. Delay propagation is therefore modeled at the airport-network scale, where temporal resolution is hourly and spatial resolution is the airport graph [6]. Excess delay is used to remove recurring calendar patterns and isolate deviations more likely to represent operational disruption.

Flight delays are difficult to forecast because they arise from interacting mechanisms operating at different spatial and temporal scales. At the local level, delays may be caused by weather, runway capacity reductions, surface congestion, gate constraints, late inbound aircraft, crew availability, or demand exceeding available airport resources. At the network level, these disruptions propagate through aircraft rotations, passenger connections, route dependencies, and airspace restrictions. Airport hierarchy and market concentration further shape the structure through which these network effects can emerge [11]. Early studies showed that airline schedules create structural channels through which delays can spread across flights and airports [12]. The importance of flight connections for delay propagation was further emphasized by Wang et al. [13]. More recent work has shown that propagated delay is a measurable system phenomenon with identifiable determinants rather than random residual variation [14–16]. These dependencies are also influenced by human factors in maintenance and inspection environments, where stress, workload, fatigue, and broader human-factor mechanisms can affect operational reliability [17–19].

Traditional delay-prediction models have relied on statistical, econometric, and conventional machine learning methods using schedule variables, historical delays, airport characteristics, weather information, and route attributes. These approaches remain valuable, especially at very short horizons where recent local observations are highly informative. Related airport surface-operation studies illustrate the importance of operational features and airport-specific context: Wang et al. [20] showed that feature importance analysis can clarify the drivers of taxi-time prediction, while Yin et al. [21] reviewed how classification, feature engineering, and modeling choices influence airport surface taxi-time analysis. However, tabular pipelines often represent airport-time observations independently, and recurrent models that flatten the airport network must infer spatial relationships indirectly from data. This can be inefficient when delay dynamics are shaped by the route topology itself. A spatial–temporal network perspective is therefore more appropriate for representing the joint evolution of airport states and network structure [22]. The same data-driven shift is visible in related aviation safety domains: BERT-based accident-report classification and socioeconomic accident analysis show how machine learning methods can extract operational risk patterns from heterogeneous aviation data [23,24], while blockchain-based record management illustrates the importance of trustworthy digital information systems for aircraft operational records [25].

Recent graph-based and generative models in broader transportation research further support the relevance of graph learning for mobility systems. Recent studies have used learning-based and spatio-temporal optimization methods for electric vehicle charg-

ing station location selection and infrastructure planning [26,27], VAE- and GAN-based architectures for multivariate road-traffic scenario generation [28], β -GCN-VAE and β -Transformer-VAE architectures for realistic urban traffic scenario generation [29], and VAE-attention-GCN models for road traffic accident severity prediction under imbalanced data [30]. Although these works address road, urban mobility, and infrastructure planning contexts rather than airport delay propagation, they demonstrate a broader methodological trend: graph neural networks and generative spatio-temporal models are increasingly used to represent structured transportation systems with heterogeneous nodes, dynamic flows, and rare-event outcomes.

Graph-based learning provides a natural framework for this problem. In an airport graph, nodes represent airports and directed edges represent traffic flows or route connectivity, allowing the model to encode the pathways through which disruptions propagate. Graph-to-sequence learning with attention has been proposed for network-wide multi-step flight delay prediction [31], and spatio-temporal propagation learning was developed to model network-wide delay dynamics [32]. Subsequent studies introduced more expressive architectures, including graph multi-attention networks for airport delay prediction [33] and self-corrective causal spatio-temporal frameworks for flight delay prediction [34]. Other recent aviation graph-learning applications address delay cause inference, holding-related delay, and terminal maneuvering area time prediction [35–37], while temporal fusion transformer forecasting has also been explored for airport delay prediction [38]. Case study evidence from Paris–Charles de Gaulle also shows that machine learning can improve off-block delay prediction when multiple operational data sources are integrated [39]. Despite these advances, many studies remain limited to select airport groups, specific regions, shorter observation periods, or targets that are difficult to compare directly across experiments.

The present study addresses this gap using a large-scale U.S. domestic flight dataset covering January 2019 to August 2023 [40]. After cleaning and filtering, the analysis includes 2,878,854 flights, 370 airport nodes, and 5334 directed route connections. Flights are aggregated into hourly airport-level bins, producing a node-time tensor that describes local departure delays, demand intensity, weather-related delay components, disruption indicators, inbound delay conditions, temporal lag features, and neighbor-based propagation signals. The regression target is the future airport-level mean departure delay in minutes, after non-negativity and clipping; excess delay relative to a train-only calendar baseline is used for disruption labels, regime analysis, and the residual XGBoost formulation. This baseline-normalized auxiliary signal separates routine schedule-driven regularity from the less predictable component of operational disruption, while reducing information leakage from validation and test periods.

The proposed model is a spatio-temporal graph convolutional network (ST-GCN) designed for multi-horizon airport-level departure delay forecasting. A six-hour input window is used to predict the next three hourly horizons. The graph convolution component mixes information across connected airports using the normalized route adjacency, while the temporal component captures short-term persistence and recovery dynamics. The framework is evaluated against two complementary baselines: an LSTM model representing a recurrent temporal learner and an XGBoost model representing a strong tabular predictor. Beyond point forecasting, this study includes rare-event classification and conformal prediction intervals because severe disruptions are infrequent but operationally important, and because decision-makers require uncertainty information in addition to mean error metrics.

The novelty of this work is empirical and operational rather than architectural. This study contributes to a large-scale national airport delay propagation framework combining

airport-level graph representation, multi-horizon forecasting, rare-disruption detection, conformal uncertainty estimation, graph-structure ablation, dynamic graph sensitivity analysis, and paired statistical evaluation. Its contribution is therefore a broader methodology for representing, predicting, and evaluating delay propagation across a national-scale airport system. First, it constructs and evaluates a substantially larger airport graph than many representative network-wide studies discussed in recent machine learning reviews of flight delay prediction [41]. Second, it integrates local airport features, temporal lags, weather-related signals, inbound conditions, and graph-neighbor information within a single hourly representation. Third, it formulates the regression target as future departure delay while using baseline-normalized excess delay to define disruption labels and regime-specific analyses. Fourth, it evaluates predictive accuracy, regime-specific errors, rare-disruption detection, conformal uncertainty calibration, model complexity, and graph sensitivity together, making this study an operational benchmark for delay propagation forecasting rather than a claim of architectural novelty.

The remainder of this paper is organized as follows. Section 2 describes the data source, cleaning procedure, temporal aggregation, route-graph construction, departure delay target, baseline-excess disruption signal, learning formulation, ST-GCN architecture, baselines, and uncertainty calibration methods. Section 3 presents the empirical results, including horizon-wise accuracy, regime-specific performance, rare-event detection, conformal intervals, and comparison with related studies. Section 4 summarizes the main findings, discusses limitations, and outlines future directions involving dynamic graph structures, richer exogenous data, and tail-risk-oriented forecasting.

2. Materials and Methods

2.1. Data Source, Cleaning, and Filtering

This study uses the Flight Delay and Cancellation Dataset (2019–2023), obtained from Kaggle and accessed in February 2026 [40]. Data cleaning removed records with invalid dates, missing origin/destination airport codes, missing scheduled departure times, canceled/diverted operations for the main delay-regression target, and non-U.S. domestic itineraries. Negative delay values were set to zero when constructing non-negative operational delay targets, and extreme delays were clipped at 180 min for model stabilization. Time windows were then generated chronologically, ensuring that input windows precede prediction windows and that train, validation, and test periods are contiguous in time. After consistency checks and the removal of records failing essential timestamp and airport-code validity constraints, the final dataset contains 2,878,854 flights spanning 1 January 2019 to 31 August 2023, as summarized in Table 1. The operational regression target is future airport-level mean departure delay in minutes, with negative values set to zero and extreme values clipped at 180 min for model stabilization. Arrival delay variables are used as inbound-context features rather than as the final regression target.

Table 1. Dataset, discretization, split, and imbalance summary.

Quantity	Value
Flights after cleaning	2,878,854
Study period	1 January 2019 to 31 August 2023
Time bin width	60 min
Total time bins	$T = 35,261$
Airports and edges	$N = 370, E = 5334$

Table 1. Cont.

Quantity	Value
Features	$F = 12$
Split (time bin indices)	train [0, 22,671), val [22,671, 30,235), test [30,235, 35,261)
Train windows (no oversampling)	22,663
Validation windows	7556
Test windows	5018
Disrupted train window fraction	$19,685/22,663 = 0.8686$
Rare window repeat factor	2
Effective train windows	42,348
Rare prevalence overall	0.008722
Rare prevalence by split	train 0.006240, val 0.012811, test 0.013768
Baseline delay mean and std	4.4407 min, 13.9502 min
Excess delay mean and positive fraction	0.82685 min, 0.03130
Flight-level delay context (mean, std)	4.138 min, 50.439 min
Flight-level delay context (median)	−6 min
Flight-level delay context (p95, p99)	69 min, 187 min
Environment	Python 3.12.7; TensorFlow 2.20.0; XGBoost 2.1.1; GPU Accelerated

2.2. Temporal Aggregation and Node-Time Tensor Construction

Flights are aggregated into fixed-width time bins of 60 min each. Let $t \in 0, 1, \dots, T - 1$ denote the discrete bin index, where $T = 35,261$ is the total number of bins. Let $i \in 1, 2, \dots, N$ denote the airport node index, where $N = 370$, which means the total number of airports in the dataset. For each time bin t and airport i , a feature vector is constructed, yielding a tensor $X \in \mathbb{R}^{T \times N \times F}$ with $F = 12$ engineered features. The node-time tensor is engineered per airport i and hourly bin t features. Each feature is constructed to capture complementary drivers of delay dynamics, including local congestion state, exogenous disruption, tail-risk intensity, short-term temporal persistence, and upstream network propagation.

The first seven features are computed directly at each airport and time bin. Feature $f = 0$ is defined as the mean departure delay of flights departing from airport i during bin t in minutes, and serves as the primary local delay state descriptor. Feature $f = 1$ is the number of departures from airport i during bin t , which operationally represents instantaneous demand and capacity pressure on runway, surface, and gate resources. Feature $f = 2$ is the mean weather-attributed delay component for departures at airport i in bin t , providing an exogenous forcing signal that helps separate meteorology-driven delay from purely endogenous congestion. Feature $f = 3$ is a binary indicator that the bin exhibits weather disruption conditions at airport i , functioning as a regime marker for non-stationary operating conditions. Feature $f = 4$ is the proportion of departures at airport i in bin t whose delay exceeds the pipeline “large-delay” threshold, serving as a tail-intensity statistic that is more sensitive to disruption onset than the mean alone. Feature $f = 5$ is the mean arrival delay for flights arriving at airport i in bin t , which captures inbound congestion and the downstream consequences of upstream disturbances; the implementation additionally applies neighbor mixing for this arrival descriptor, which increases sensitivity to propagated conditions. Feature $f = 6$ is the fraction of arrivals into airport i in bin t exceeding the large-delay threshold, which is similarly influenced by neighbor mixing and provides a propagation-aware measure of inbound tail risk.

Short-term persistence and recovery are encoded by three explicit temporal lag features derived from the mean departure delay. Features $f = 7$, $f = 8$, and $f = 9$ are defined, respectively, as $\text{mean_dep_delay}_{i,t-1}$, $\text{mean_dep_delay}_{i,t-2}$, and $\text{mean_dep_delay}_{i,t-3}$. These lagged signals represent the empirically strong autocorrelation of airport delay states over 1–3 h windows, capturing backlog dissipation, rolling resource constraints, and tactical flow-management persistence.

Finally, two graph-propagation features inject upstream delay information using the normalized route adjacency. Feature $f = 10$ is $\text{neighbour_dep_delay_prev}_{i,t}$ (minutes), defined as the graph-averaged neighbor departure delay at the previous bin, computed by aggregating $\text{mean_dep_delay}_{j,t-1}$ over connected airports j using the normalized adjacency weights. Feature $f = 11$ is $\text{neighbour_dep_share_big_prev}_{i,t}$ (unitless fraction), defined analogously by aggregating $\text{dep_share_big}_{j,t-1}$ over neighbors. These propagation features provide a mechanistic proxy for delay diffusion along the airport network, enabling the model to detect upstream disruption signatures before they fully manifest locally.

Additional operational variables that would enrich the outcome, including aircraft rotation continuity, gate-level turnaround time, runway configuration, airport acceptance rates, and airspace flow restrictions, were not consistently available in the public dataset. The model therefore uses observable proxies, including departure demand, inbound delay pressure, weather-attributed delay, arrival-tail intensity, route distance, hub/connectivity indicators, and graph-neighbor delay pressure. Taxi-time and cancellation/diversion fields are available in the broader source data, but canceled/diverted operations are excluded from the main delay-regression target and taxi variables are not used as direct target variables. These variables should be interpreted as predictive operational descriptors rather than causal measurements of the underlying mechanisms.

$$X_{t,i,f} = \text{Agg}_f(S_t, i) \quad (1)$$

In Equation (1), $X_{t,i,f}$ is the f -th feature at time bin t and airport i ; $S_{t,i}$ denotes the set of flights mapped to bin t and airport i under the aggregation rules; and $\text{Agg}_f(\cdot)$ is the feature-specific aggregation operator. Feature engineering includes airport-level operational load, delay, and disruption indicators at hourly resolution in function of the list of the 12 features. The chronological split is defined over time bins as train $[0, 22,671)$, validation $[22,671, 30,235)$, and test $[30,235, 35,261)$; this is mapped in Table 1.

2.3. Static Route Graph Construction and Normalization

A static directed route graph is constructed from the observed traffic flows. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ encodes directed connectivity from origin airport i to destination airport j , weighted by route intensity (e.g., route flight counts) and filtered by a minimum-flight threshold.

$$A_{ij} = c_{ij} I(c_{ij} \geq \gamma) \quad (2)$$

In Equation (2), A_{ij} is the directed edge weight from node i to node j ; c_{ij} is the observed route count between airports i and j over the study period; $I(\cdot)$ is the indicator function; and γ is the minimum-flight threshold. The resulting graph contains $E = 5334$ directed edges. Graph convolution uses a symmetrically normalized adjacency with self-loops.

$$\tilde{A} = A + I \quad (3)$$

In Equation (3), $\tilde{A} \in \mathbb{R}^{N \times N}$ is the adjacency with self-loops; $I \in \mathbb{R}^{N \times N}$ is the identity matrix.

$$D_{ii} = \sum_{j=1}^N \tilde{A}_{ij} \quad (4)$$

In Equation (4), $D \in \mathbb{R}^{N \times N}$ is a diagonal degree matrix whose diagonal entry D_{ii} is the out-degree (row-sum) of node i under $\tilde{A}$.

$$A_{\text{norm}} = D^{-1/2} \tilde{A} D^{-1/2} \quad (5)$$

In Equation (5), $A_{\text{norm}} \in \mathbb{R}^{N \times N}$ is the symmetrically normalized adjacency; $D^{-1/2}$ is the diagonal matrix with entries $D_{ii}^{-1/2}$.

2.4. Baseline Delay and Excess Delay Definition

To isolate network-driven delay propagation from predictable schedule-driven variation, a baseline delay is estimated using train-only calendar strata defined by airport, hour-of-day, day-of-week, and month. Let $b_{i,t}$ denote the baseline at airport i and time bin t .

$$b_{i,t} = E(y_{i,t} \mid i, h(t), d(t), m(t), t \in \mathcal{T}_{\text{train}}) \quad (6)$$

In Equation (6), $y_{i,t}$ is the observed mean departure delay in minutes, aggregated at airport i and time t under the target construction; $h(t)$ is hour-of-day, $d(t)$ is day-of-week, and $m(t)$ is month; and $\mathcal{T}_{\text{train}}$ denotes the set of time bins in the training split. The baseline distribution has mean 4.4407 min and standard deviation 13.9502 min, and the derived excess departure delay has mean 0.82685 min with fraction positive 0.03130, given in Table 1. The clipping and log1p transformation are used for empirical stabilization under heavy-tailed delay distributions. Clipping at 180 min reduces the influence of extreme outliers that are rare, operationally heterogeneous, and often driven by exogenous disruption mechanisms. The log1p transformation compresses the right tail during optimization, while all reported RMSE, MAE, regime errors, and plots are computed after inverse transformation back to minutes. Therefore, the operational interpretation remains in the original minute scale.

$$y^{\text{excess}}_{i,t} = \text{clip}(y_{i,t} - b_{i,t}, 0, y_{\text{max}}) \quad (7)$$

In Equation (7), $y^{\text{excess}}_{i,t}$ is excess departure delay in minutes at airport i and time t ; y_{max} is a maximum cap (minutes) defined in the implementation; and $\text{clip}(u, a, b)$ returns $\min(\max(u, a), b)$. A rare disruption label is defined by thresholding the excess delay.

$$y^{\text{rare}}_{i,t} = I(y^{\text{excess}}_{i,t} \geq \theta) \quad (8)$$

In Equation (8), $y^{\text{rare}}_{i,t} \in 0, 1$ is the rare-event indicator at airport i and time t , and θ is the disruption threshold in minutes. The rare/disruption prevalence is 0.008722 overall, with train 0.006240, validation 0.012811, and test 0.013768, indicating a distribution shift toward higher disruption prevalence in later periods, as given in Table 1.

2.5. Learning Problem Definition (Multi-Horizon, Multi-Task)

Learning is posed as multi-horizon forecasting with $t_{\text{in}} = 6$ context bins and $t_{\text{out}} = 3$ forecast bins. For each sample indexed by anchor time t , the input context comprises the preceding window of node features, and targets comprise future airport-level mean departure delay and rare-event indicators derived from excess departure delay. The implementation constructs training windows using a contiguous segment length $t_{\text{in}} + t_{\text{out}} - 1 = 8$ for alignment, consistent with the tensor shape. The regression target is the future airport-level departure delay in minutes, computed after non-negativity and clipping; neural network regression is performed in log space to reduce tail dominance and stabilize gradients.

Because the observations form a temporally ordered spatio-temporal sequence, the train, validation, and test partitions were kept contiguous in time rather than randomly sampled. This design prevents direct future-to-past leakage in preprocessing, model

selection, and evaluation. Nevertheless, temporal autocorrelation remains relevant because daily, weekly, and seasonal operating patterns recur across years. The purpose of the chronological split is therefore not to eliminate all temporal dependence, which is intrinsic to the forecasting problem, but to ensure that all learned parameters, scaling statistics, baseline-delay estimates, and hyperparameter choices are estimated only from information available before the evaluation period. The train-only calendar baseline and train-only feature scaling are used specifically to avoid validation/test information entering the model through preprocessing.

$$z^{\text{reg}}_{h,i,t} = \log(1 + y^{\text{dep}}_{i,t+h}) \quad (9)$$

In Equation (9), $z^{\text{reg}}_{h,i,t}$ is the regression target in log-space; $h \in 1, 2, 3$ is the forecast horizon in bins; and $y^{\text{dep}}_{i,t+h}$ is the future airport-level mean departure delay in minutes after non-negativity and clipping. Predictions are mapped back to minutes by the inverse transform.

$$\hat{y}^{\text{dep}}_{h,i,t} = \exp(\hat{z}^{\text{reg}}_{h,i,t}) - 1 \quad (10)$$

In Equation (10), $\hat{z}^{\text{reg}}_{h,i,t}$ is the model output in log-space and $\hat{y}^{\text{dep}}_{h,i,t}$ is the predicted mean departure delay in minutes. A parallel classification head predicts the probability of rare disruption via a sigmoid link function.

$$\hat{p}_{h,i,t} = \frac{1}{1 + \exp(-\hat{z}^{\text{cls}}_{h,i,t})} \quad (11)$$

In Equation (11), $\hat{z}^{\text{cls}}_{h,i,t}$ is the classification logit and $\hat{p}_{h,i,t} \in (0, 1)$ is the predicted probability that $y^{\text{rare}}_{i,t+h} = 1$.

2.6. ST-GCN Architecture

The main model uses a static route-count adjacency to represent the persistent structural connectivity of the U.S. airport network. To address the limitation of a purely static topology, an hourly dynamic graph sensitivity analysis is added in which an adjacency matrix is constructed from the observed origin–destination traffic intensity within each time bin. The dynamic adjacency incorporates hourly departure and arrival frequencies, route-level connection intensity, and peak/off-peak operational variation. This is reported as a sensitivity and ablation analysis because the static graph represents long-run route availability whereas the dynamic graph represents short-term operational activation. The implemented dynamic variant uses dynamic graph-derived features and a time-averaged normalized dynamic adjacency; it is not presented as a fully learnable time-varying graph-convolution operator. The selected model uses three stacked spatio-temporal blocks and a dual-head output, with configuration selected by validation RMSE in log-space. The best configuration is $gcn_units = 96$, $temporal_units = 64$, dropout 0.2, and learning rate 10^{-3} , given in Table 2, and the corresponding parameter count is 70,058, given in Table 3. Figure 1 summarizes the data flow from the flight records through hourly aggregation, graph-aware ST-GCN encoding, node gating, dual prediction heads, and evaluation outputs.

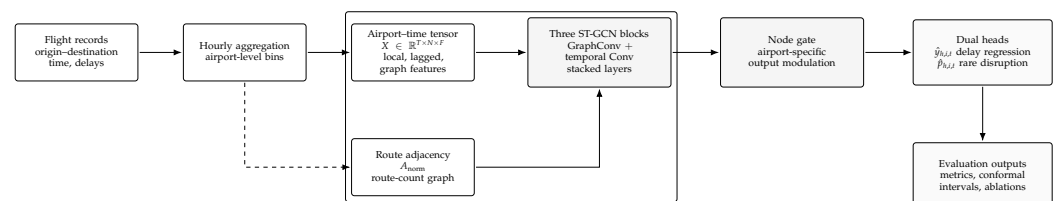


Figure 1. Architecture of the proposed airport delay ST-GCN forecasting framework.

Table 2. Model hyperparameters and selection.

Model	Configuration	Key Units	Dropout	Learning Rate
ST-GCN	cfg1	gcn 64, temporal 64	0.2	10^{-3}
ST-GCN	cfg2	gcn 96, temporal 64	0.2	10^{-3}
ST-GCN	cfg3	gcn 64, temporal 96	0.3	5×10^{-4}
LSTM	baseline	hidden 64	0.0	10^{-3}
XGBoost	direct ($h = 1, 2, 3$)	depth 8, min_child_weight 5, subsample 0.9, colsample 0.9, $\lambda = 1.0$	–	0.03

Table 3. Model complexity in trainable parameters. ST-GCN values correspond to the tested configurations reported by the model summaries.

Model	Trainable Parameters
ST-GCN cfg1	47,146
ST-GCN cfg2	70,058
ST-GCN cfg3	69,962
LSTM baseline	1,225,430

Figure 1 clarifies that the article applies ST-GCN as one component of a complete airport delay propagation pipeline rather than as a newly proposed graph-convolution operator. The contribution lies in the end-to-end construction and evaluation of the national airport graph, hourly node-time tensor, graph-aware forecasting blocks, node gate, dual regression/classification outputs, and downstream metrics, conformal intervals, and ablations.

Graph convolution at each time step is defined as

$$H^{(\ell+1)}t = \text{ReLU}(\text{Anorm}H^{(\ell)}tW^{(\ell)}) \quad (12)$$

Regarding Table 2, the dropout for the LSTM baseline is 0.0 because no dropout argument is specified in the LSTM layer constructor in the implementation; Keras defaults are used in that case. XGBoost learning rate corresponds to the eta parameter.

In Equation (12), $H^{(\ell)}t \in \mathbb{R}^{N \times C_\ell}$ is the node embedding matrix at time index t and layer ℓ ; C_ℓ is the channel dimension; $W^{(\ell)} \in \mathbb{R}^{C_\ell \times C_{\ell+1}}$ is a trainable weight matrix; and $\text{ReLU}(u) = \max(0, u)$ is applied elementwise. Temporal modeling within each ST block is implemented as a causal temporal convolution applied along the time axis.

$$U_t = \sum_{k=0}^{K-1} H_{t-k}V_k \quad (13)$$

In Equation (13), $U_t \in \mathbb{R}^{N \times C}$ is the temporally filtered embedding at time index t ; K is the temporal kernel length; and $V_k \in \mathbb{R}^{C \times C}$ are trainable temporal convolution weights. The combination of Equations (12) and (13) yields a joint spatio-temporal representation suitable for delay propagation dynamics. The architecture includes a node-gating mechanism to modulate airport-specific output sensitivity after the shared spatio-temporal representation has been learned. The purpose of this mechanism is not to introduce a new graph-convolution operator but to allow the regression and classification heads to adapt to airport heterogeneity. Airports differ substantially in traffic volume, hub status, route connectivity, operational buffers, weather exposure, and disruption propagation relevance. A single shared output scale may therefore under-represent highly connected hubs or over-represent small airports with sparse observations. The node gate learns a normalized airport-specific weight that adjusts horizon-wise outputs according to the relative contribution of each node. To verify that this mechanism is not merely cosmetic, the revised

experiments include an ablation without node gating and compare its performance with the gated ST-GCN. The learned gate weights are also compared with airport connectivity and delay-regime statistics to assess whether the mechanism captures meaningful operational heterogeneity. A softmax gate over airports is defined as

$$g_i = \frac{\exp(a_i)}{\sum_{j=1}^N \exp(a_j)} \quad (14)$$

In Equation (14), $a_i \in \mathbb{R}$ is a trainable gating logit for node i and $g_i \in (0, 1)$ is the normalized gate weight, with $\sum_{i=1}^N g_i = 1$. The gating is applied multiplicatively to horizon-wise outputs:

$$\tilde{o}_{h,i,t} = g_i o_{h,i,t} \quad (15)$$

In Equation (15), $o_{h,i,t}$ is a pre-gated output for horizon h , node i , and anchor time t , and $\tilde{o}_{h,i,t}$ is the gated output. Separate gates are used for regression and classification heads, consistent with the distinct node-gate layers reported in the model summaries. The regression head produces $\hat{z}^{\text{reg}}_{h,i,t}$ and the classification head produces $\hat{z}^{\text{cls}}_{h,i,t}$ via linear projections from the final spatio-temporal embedding S_t .

$$\hat{z}^{\text{reg}}_{h,i,t} = w^{\text{reg}}_h \cdot S_{t,i} + b^{\text{reg}}_h \quad (16)$$

In Equation (16), $S_{t,i} \in \mathbb{R}^C$ is the final embedding at node i and time t ; $w^{\text{reg}}_h \in \mathbb{R}^C$ and $b^{\text{reg}}_h \in \mathbb{R}$ are horizon-specific regression parameters. An analogous expression holds for $\hat{z}^{\text{cls}}_{h,i,t}$ with classification parameters. The probability output $\hat{p}_{h,i,t}$ is computed via Equation (11), and the regression output in minutes is computed via Equation (10).

2.7. Loss Function and Optimization

Training minimizes a weighted multi-task objective combining regression error in log-space and binary cross-entropy for rare-event prediction.

$$L_{\text{reg}} = \frac{1}{M} \sum_{m=1}^M \sum_{h=1}^{t_{\text{out}}} \sum_{i=1}^N (\hat{z}^{\text{reg}}_{h,i,t_m} - z^{\text{reg}}_{h,i,t_m})^2 \quad (17)$$

In Equation (17), M is the number of training windows, t_m is the anchor time for window m , and z^{reg}_{h,i,t_m} is defined in Equation (9).

$$L_{\text{cls}} = -\frac{1}{M} \sum_{m=1}^M \sum_{h=1}^{t_{\text{out}}} \sum_{i=1}^N (y^{\text{rare}}_{i,t_m} + h \log(\hat{p}_{h,i,t_m}) + (1 - y^{\text{rare}}_{i,t_m} + h) \log(1 - \hat{p}_{h,i,t_m})) \quad (18)$$

In Equation (18), $y^{\text{rare}}_{i,t_m} + h$ is defined in (8) and $\hat{p}_{h,i,t_m}$ is defined in (11).

$$L = \alpha L_{\text{reg}} + \beta L_{\text{cls}} \quad (19)$$

In Equation (19), α and β are non-negative scalar weights set in the implementation to balance regression fidelity and rare-event sensitivity. Optimization uses Adam, with learning rates given in Table 2, dropout as specified per configuration, a maximum of 80 epochs, and early stopping with patience based on validation RMSE in log-space.

Window generation uses $t_{\text{in}} = 6$ and $t_{\text{out}} = 3$, yielding 22,663 train windows, 7556 validation windows, and 5018 test windows. To counter class imbalance, disrupted windows are repeated with a factor of 2, resulting in 42,348 effective training windows. The disrupted fraction among training windows is $19,685/22,663 = 0.8686$ under the disruption-window definition used in the run manifest.

An additional validation-only optimization search was conducted to test whether the global and severe-delay errors could be reduced without using the final test period for model selection. The search cached the expensive preprocessing outputs, sampled complete forecast windows rather than individual airport observations, and evaluated candidate losses, target transformations, post hoc calibration rules, and XGBoost objectives using a composite validation score that balanced global RMSE/MAE with short-delay and severe-delay errors.

2.8. Training Dynamics and Convergence Behavior

The optimization behavior of the proposed spatio-temporal graph convolutional network was monitored during training in order to verify numerical stability, convergence speed, and generalization behavior. Figure 2 illustrates the evolution of the training loss and validation error across training epochs. This figure shows the evolution of the balanced log-MAE training loss and the validation RMSE computed in log space. The validation error decreases rapidly during early epochs and reaches a minimum around epochs 8–9, after which a gradual increase indicates the onset of mild overfitting.

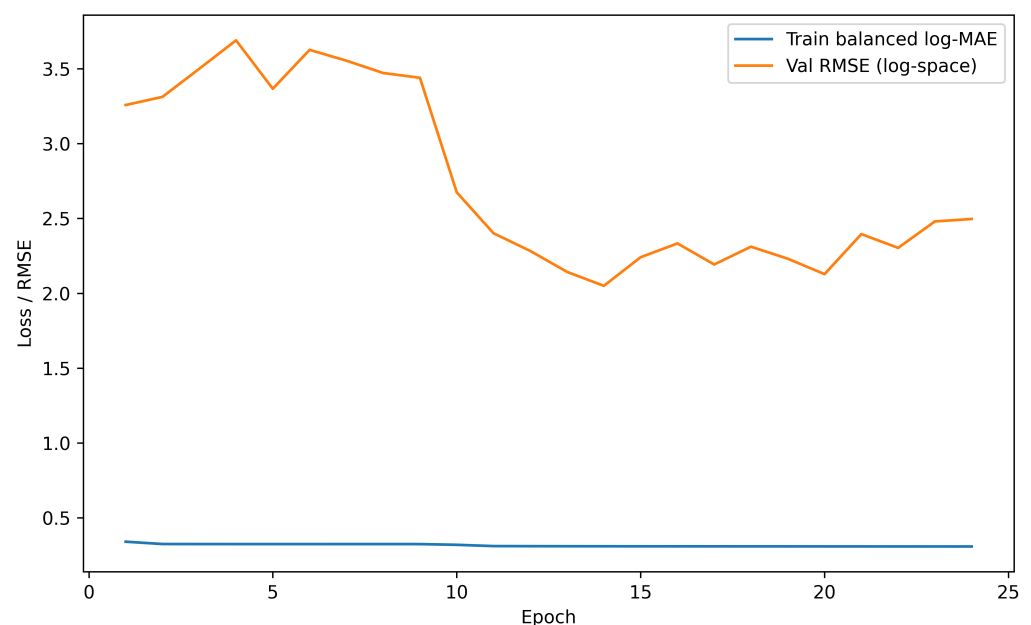


Figure 2. Training dynamics of the ST-GCN model.

The training loss corresponds to the balanced log-MAE objective defined in the previous subsection, which was designed to mitigate the influence of the heavy-tailed distribution of flight delays. As shown in Figure 2, the training loss exhibits a stable and slowly decreasing trajectory throughout the optimization process. This behavior indicates that gradient updates remain numerically well behaved and that the loss function provides a smooth optimization landscape. More importantly, the validation RMSE in log space decreases substantially during the early stages of training. This rapid improvement suggests that the model quickly learns the dominant delay propagation patterns present in the spatio-temporal airport network. The validation error reaches its minimum around epochs 8–9, corresponding to the point of best generalization performance on unseen data.

After this point, the validation error begins to increase gradually while the training loss continues to decrease slightly. This divergence is characteristic of mild overfitting and is commonly observed in deep learning models once the model begins to adapt excessively to training-specific noise. To mitigate this effect, model selection is based on the epoch that achieves the lowest validation error rather than the final training epoch.

The observed convergence behavior confirms three desirable properties of the proposed learning framework. First, the optimization process remains numerically stable, with no evidence of exploding gradients or oscillatory behavior. Second, the model converges rapidly, reaching its optimal validation performance within fewer than ten epochs. Third, the presence of a well-defined validation minimum enables reliable early-stopping and model checkpoint selection.

2.9. Baselines

A temporal LSTM baseline is trained on flattened node features, mapping each time step to a vector in $\mathbb{R}^{NF}$ and processing sequences with an LSTM layer of 64 hidden units followed by a dense projection to $t_{\text{out}} \times N$ outputs. The LSTM model has 1,225,430 parameters, as per Table 3. The XGBoost baselines are trained as direct per-horizon tabular regressors for $h = 1$, $h = 2$, and $h = 3$, aligned with the same forecast windows used by the neural models. Two feature sets are evaluated: XGBoost with temporal features, using node identity, calendar variables, baseline delay, and lagged local airport features; and XGBoost with spatial features, additionally using graph-neighbor pressure, node degree, weighted in-degree and out-degree, hub status, and route distance information where available.

2.10. Uncertainty and Calibration

Uncertainty quantification uses split-conformal prediction on absolute residuals in real minutes, computed on a calibration set and applied to test predictions.

$$r_n = |y_n^{\text{dep}} - \hat{y}_n^{\text{dep}}| \quad (20)$$

In Equation (20), y_n^{dep} is the true departure delay in minutes for calibration example n and $\hat{y}_n^{\text{dep}}$ is the corresponding prediction in minutes.

$$q_{1-\delta} = \text{Quantile } 1 - \delta(r_n, n = 1^{N_{\text{cal}}}) \quad (21)$$

In Equation (21), $q_{1-\delta}$ is the $(1 - \delta)$ quantile of calibration residuals; N_{cal} is the number of calibration examples; and $\delta \in 0.10, 0.05$ corresponds to 90% and 95% nominal coverage.

$$\mathcal{I}\delta(x) = [\hat{y}^{\text{dep}}(x) - q_{1-\delta}, \hat{y}^{\text{dep}}(x) + q_{1-\delta}] \quad (22)$$

In Equation (22), $\mathcal{I}\delta(x)$ is the conformal prediction interval in minutes for input x . The reported absolute error quantiles are $q_{0.90} = 0.00$ min and $q_{0.95} = 6.72$ min, with empirical test coverages approximately 89.8% and 94.6% for the 90% and 95% intervals. Probability calibration uses temperature scaling on classification logits. Split-conformal prediction provides marginal finite-sample coverage under the assumption that calibration and deployment examples are exchangeable. This assumption is strong in transportation systems because airport operations are affected by seasonality, demand shocks, weather regimes, and changes in disruption prevalence. In the present dataset, the validation and test periods show a higher disruption prevalence than the training period, indicating a measurable distribution shift. For this reason, the conformal intervals are interpreted as calibration-based uncertainty estimates rather than unconditional guarantees under arbitrary non-stationarity. We therefore report empirical test coverage alongside nominal coverage and treat deviations from nominal coverage as evidence about calibration robustness under the observed temporal shift.

$$\hat{p}^{(T)}_{h,i,t} = \frac{1}{1 + \exp(-\hat{z}^{\text{cls}}_{h,i,t}/T)} \quad (23)$$

In Equation (23), $T > 0$ is the temperature parameter and $\hat{p}_{h,i,t}^{(T)}$ is the calibrated probability. The learned temperature is $T^* = 1.200$, with validation AUC 0.8670 and validation Brier score 0.0122 before and after scaling, indicating negligible change under this calibration step. Decision threshold selection is performed on the validation set and not on the test set. The validation F_1 score is reported as a threshold-selection criterion because it balances precision and recall under severe class imbalance; however, this criterion is not intended to represent a universal operational optimum. In practice, airport disruption detection is cost-asymmetric: a missed severe disruption can propagate through gates, crews, aircraft rotations, passenger connections, and air traffic flow restrictions, whereas a false alarm mainly consumes monitoring or planning resources. The operating threshold should therefore be interpreted as a tunable decision parameter. If C_{FN} and C_{FP} denote the operational costs of missed detections and false alarms, respectively, an operator may select the threshold that minimizes the expected validation cost rather than the threshold that maximizes F_1 .

$$F_1(\tau) = \frac{2 \text{Precision}(\tau) \text{Recall}(\tau)}{\text{Precision}(\tau) + \text{Recall}(\tau)} \quad (24)$$

In Equation (24), $\tau \in (0, 1)$ is a probability threshold applied to $\hat{p}^{(T)}$; precision is the fraction of predicted positives that are true positives; and recall is the fraction of true positives that are detected. The selected one-hour ST-GCN threshold is $\tau^* = 0.125$ with validation $F_1 = 0.193$, selected on the validation set only.

In the validation-selected set, the promoted ST-GCN setting uses the second configuration in Table 3, with 70,058 trainable parameters. This remains substantially smaller than the LSTM baseline and preserves the parameter-efficiency interpretation while reflecting the final optimized run.

3. Results and Discussion

3.1. Primary Accuracy Across Horizons

Primary regression accuracy is reported in minutes after applying the inverse transform using Equation (10). Table 4 summarizes the RMSE and MAE for the aligned final evaluation over horizons $h \in \{1, 2, 3\}$. The XGBoost baselines are no longer evaluated only at $h = 1$; instead, separate direct regressors are trained for each horizon using either temporal-only features or the extended spatial-feature set.

Table 4 shows that, among the primary point-forecasting models, LSTM gives the lowest aligned point-forecast errors across the three horizons, with RMSE between 11.2859 and 11.3691 min and MAE between 1.7539 and 1.7627 min. Therefore, the contribution of ST-GCN should not be interpreted as universal point-forecast superiority. Its role in this study is to provide an explicit graph-aware representation of airport delay propagation, allowing airport connectivity, graph ablations, dynamic graph sensitivity, node-level outputs, and propagation-aware severe-delay analysis to be evaluated within a single multi-horizon framework. In the final aligned multi-horizon comparison, both XGBoost variants are materially worse than the neural models, and adding spatial graph features to XGBoost does not materially improve point accuracy relative to the temporal-only feature set.

Table 5 adds the full-test metrics for the hourly dynamic graph sensitivity and the hybrid ST-GCN–XGBoost output. The dynamic-hourly ST-GCN improves RMSE relative to the observed static ST-GCN at all three horizons, but it does not improve MAE, which confirms that time-varying route activation affects squared-error behavior more than the dominant near-zero absolute-error mass. The hybrid output reduces MAE relative to XGBoost alone, but it remains less accurate than the neural models in global RMSE and MAE.

Table 4. Forecast accuracy by model and horizon under the aligned final evaluation.

Model	Horizon h	RMSE (min)	MAE (min)	Test Samples
ST-GCN, observed graph	1	11.5141	2.0513	1,856,660
ST-GCN, observed graph	2	11.6043	1.7795	1,856,660
ST-GCN, observed graph	3	11.6028	1.7790	1,856,660
LSTM	1	11.3691	1.7556	1,856,660
LSTM	2	11.2859	1.7539	1,856,660
LSTM	3	11.3429	1.7627	1,856,660
XGBoost, temporal features	1	17.6896	5.4273	1,856,660
XGBoost, temporal features	2	17.6996	5.4291	1,856,660
XGBoost, temporal features	3	17.7083	5.4307	1,856,660
XGBoost, spatial features	1	17.6881	5.4277	1,856,660
XGBoost, spatial features	2	17.6986	5.4295	1,856,660
XGBoost, spatial features	3	17.7079	5.4312	1,856,660

Table 5. Additional final-test metrics, including the dynamic graph sensitivity and hybrid outputs.

Model	Horizon h	RMSE (min)	MAE (min)	Test Samples
ST-GCN, observed graph	1	11.5141	2.0513	1,856,660
ST-GCN, observed graph	2	11.6043	1.7795	1,856,660
ST-GCN, observed graph	3	11.6028	1.7790	1,856,660
ST-GCN, hourly dynamic graph	1	11.4086	1.9006	1,856,660
ST-GCN, hourly dynamic graph	2	11.4730	1.9247	1,856,660
ST-GCN, hourly dynamic graph	3	11.4238	1.9028	1,856,660
LSTM	1	11.3691	1.7556	1,856,660
LSTM	2	11.2859	1.7539	1,856,660
LSTM	3	11.3429	1.7627	1,856,660
XGBoost, temporal features	1	17.6896	5.4273	1,856,660
XGBoost, temporal features	2	17.6996	5.4291	1,856,660
XGBoost, temporal features	3	17.7083	5.4307	1,856,660
XGBoost, spatial features	1	17.6881	5.4277	1,856,660
XGBoost, spatial features	2	17.6986	5.4295	1,856,660
XGBoost, spatial features	3	17.7079	5.4312	1,856,660
Hybrid ST-GCN + XGBoost	1	17.7859	4.4794	1,856,660
Hybrid ST-GCN + XGBoost	2	17.8180	4.3207	1,856,660
Hybrid ST-GCN + XGBoost	3	17.8306	4.3232	1,856,660

Table 6 indicates that graph structure affects the forecast, but not as a uniform point-accuracy advantage for the observed static adjacency. The hourly dynamic graph sensitivity gives a lower RMSE than the observed static ST-GCN at all three horizons, whereas the no-node-gate variant gives the lowest RMSE at $h = 2$ and $h = 3$. The MAE pattern is more mixed because several simplified variants stay closer to the dominant near-zero delay regime. These results support the reviewer concern that a static route graph is incomplete: time-varying route activation changes the squared-error behavior, but the implemented dynamic variant remains a sensitivity analysis based on dynamic features and a time-averaged adjacency rather than a fully learnable dynamic graph convolution. The node-gating ablation is therefore interpreted cautiously as a lightweight airport-heterogeneity layer, not as a source of universal accuracy improvement.

Table 7 shows that the dynamic-hourly graph mainly changes route activation rather than the average nonzero edge weight. Peak periods activate approximately 105 routes per hour compared with 68 during off-peak periods, and morning peak has the highest route activation. Across all hours, the dynamic active routes have high overlap with the static route set (mean overlap rate 0.9853), but only a small fraction of the static route graph is active in a typical hour (mean static coverage 0.0149), with high hour-to-hour active-route turnover (0.9873). The dynamic-hourly ST-GCN variant in Table 6 should therefore be

interpreted as a dynamic-feature and time-averaged-adjacency sensitivity analysis, not as a fully learnable time-varying graph-convolution model.

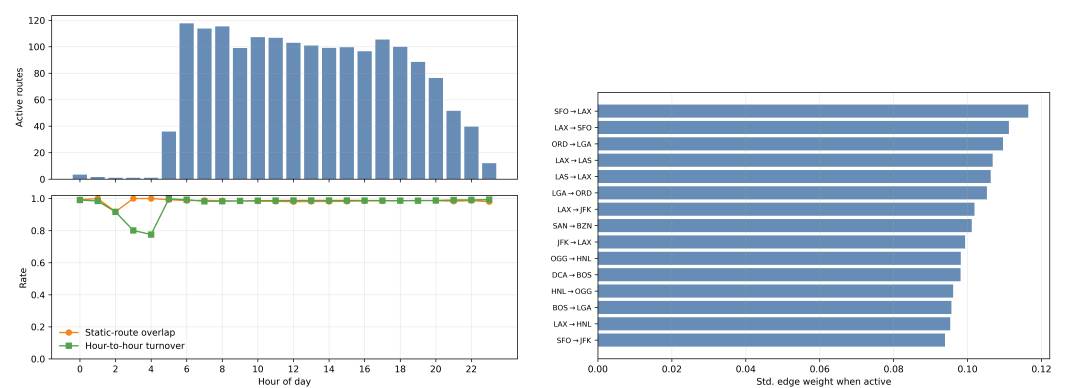
Table 6. Graph-structure, spatial-feature, and node-gating ablations by forecast horizon.

Model	RMSE, $h = 1$	MAE, $h = 1$	RMSE, $h = 2$	MAE, $h = 2$	RMSE, $h = 3$	MAE, $h = 3$
ST-GCN, observed graph	11.5141	2.0513	11.6043	1.7795	11.6028	1.7790
ST-GCN, identity graph	11.9199	2.0034	11.9894	2.0359	11.7063	1.9533
ST-GCN, randomized graph	11.6056	1.7799	11.6043	1.7795	11.6028	1.7790
ST-GCN, hourly dynamic graph	11.4086	1.9006	11.4730	1.9247	11.4238	1.9028
ST-GCN, no graph-neighbor features	11.6056	1.7799	11.5632	2.0434	11.5910	2.0745
ST-GCN, no node gate	11.6056	1.7799	11.3448	1.8561	11.3277	1.8074
XGBoost, temporal features	17.6896	5.4273	17.6996	5.4291	17.7083	5.4307
XGBoost, spatial features	17.6881	5.4277	17.6986	5.4295	17.7079	5.4312

Table 7. Hourly dynamic graph summary by operating period.

Period	Active Routes	Mean Edge Weight	Graph Density	Departures/Arrivals
Peak	105.34 $\pm$ 29.05	0.6975 $\pm$ 0.0043	0.000772	106.56
Off-peak	68.28 $\pm$ 43.93	0.6968 $\pm$ 0.0132	0.000500	68.91
Morning peak	111.48 $\pm$ 28.74	0.6978 $\pm$ 0.0043	0.000817	112.85
Midday	102.80 $\pm$ 23.27	0.6968 $\pm$ 0.0040	0.000753	103.78
Evening peak	93.38 $\pm$ 27.06	0.6969 $\pm$ 0.0042	0.000684	94.31
Night	25.07 $\pm$ 24.23	0.6969 $\pm$ 0.0199	0.000184	25.26

Figure 3 provides the visual support for the dynamic graph analysis in Table 7. The left panel shows that route activation follows a pronounced time-of-day pattern, while the right panel identifies routes whose activation varies most strongly over time. This strengthens the interpretation that a static route graph captures long-run connectivity but cannot fully describe hourly propagation opportunities.



(a) Hourly route activation.

(b) Routes with strongest temporal variation.

Figure 3. Dynamic graph evidence from hourly origin–destination activation.

Table 8 reports uncertainty by resampling forecast windows rather than flattened airport-node observations. The intervals confirm that the neural models remain separated from the XGBoost baselines in the aligned run. The ST-GCN and LSTM intervals overlap in RMSE, but their MAE intervals are separated, consistent with LSTM's stronger performance in the near-zero regime and with the interpretation that ST-GCN is primarily a graph-aware propagation model rather than the universal point-forecast winner.

Table 9 shows that both neural models have significantly lower window-level MAE values than XGBoost with spatial features at all horizons. It also confirms that LSTM has a significantly lower MAE than the observed-graph ST-GCN. Within the ST-GCN ablations, paired MAE tests sometimes favor simpler or ungated variants, even though RMSE trends differ in Table 6; therefore, small architectural differences should be interpreted with metric-specific caution rather than as uniform superiority.

Table 8. Bootstrap confidence intervals over forecast windows for RMSE and MAE.

Model	h	RMSE	RMSE 95% CI	MAE	MAE 95% CI
ST-GCN, observed graph	1	11.5141	[11.3390, 11.6933]	2.0513	[2.0070, 2.0978]
ST-GCN, observed graph	2	11.6043	[11.4418, 11.7649]	1.7795	[1.7440, 1.8175]
ST-GCN, observed graph	3	11.6028	[11.4325, 11.7690]	1.7790	[1.7402, 1.8176]
LSTM	1	11.3691	[11.2054, 11.5366]	1.7556	[1.7192, 1.7917]
LSTM	2	11.2859	[11.1275, 11.4460]	1.7539	[1.7204, 1.7897]
LSTM	3	11.3429	[11.1755, 11.5056]	1.7627	[1.7264, 1.7988]
XGBoost, temporal features	1	17.6896	[17.5226, 17.8563]	5.4273	[5.3543, 5.4975]
XGBoost, temporal features	2	17.6996	[17.5406, 17.8763]	5.4291	[5.3600, 5.5044]
XGBoost, temporal features	3	17.7083	[17.5341, 17.8747]	5.4307	[5.3564, 5.5046]
XGBoost, spatial features	1	17.6881	[17.5210, 17.8552]	5.4277	[5.3548, 5.4979]
XGBoost, spatial features	2	17.6986	[17.5397, 17.8752]	5.4295	[5.3604, 5.5051]
XGBoost, spatial features	3	17.7079	[17.5334, 17.8740]	5.4312	[5.3570, 5.5050]

Table 9. Paired forecast-window tests using MAE loss. Negative Δ MAE means model A has lower window-level MAE than model B.

Comparison (A–B)	h	Δ MAE	Wilcoxon p	DM-Style Statistic/ p
ST-GCN, observed graph–XGBoost, spatial features	1	−3.3764	0.0000	−140.083/0.0000
ST-GCN, observed graph–XGBoost, spatial features	2	−3.6500	0.0000	−114.533/0.0000
ST-GCN, observed graph–XGBoost, spatial features	3	−3.6522	0.0000	−97.898/0.0000
LSTM–XGBoost, spatial features	1	−3.6720	0.0000	−150.534/0.0000
LSTM–XGBoost, spatial features	2	−3.6756	0.0000	−112.993/0.0000
LSTM–XGBoost, spatial features	3	−3.6686	0.0000	−95.415/0.0000
ST-GCN, observed graph–LSTM	1	0.2957	0.0000	34.535/2.434 $\times 10^{-261}$
ST-GCN, observed graph–LSTM	2	0.0256	1.655 $\times 10^{-41}$	12.688/6.853 $\times 10^{-37}$
ST-GCN, observed graph–LSTM	3	0.0164	2.369 $\times 10^{-31}$	7.146/8.922 $\times 10^{-13}$
ST-GCN, observed graph–ST-GCN, identity graph	1	0.0479	0.2476	5.259/1.448 $\times 10^{-7}$
ST-GCN, observed graph–ST-GCN, identity graph	2	−0.2564	0.0000	−27.097/1.054 $\times 10^{-161}$
ST-GCN, observed graph–ST-GCN, identity graph	3	−0.1743	1.565 $\times 10^{-116}$	−17.788/8.791 $\times 10^{-71}$
ST-GCN, observed graph–ST-GCN, randomized graph	1	0.2714	0.0000	32.660/5.760 $\times 10^{-234}$
ST-GCN, observed graph–ST-GCN, randomized graph	2	0.0000	–	identical losses
ST-GCN, observed graph–ST-GCN, randomized graph	3	0.0000	–	identical losses
ST-GCN, observed graph–ST-GCN, no node gate	1	0.2714	0.0000	32.660/5.760 $\times 10^{-234}$
ST-GCN, observed graph–ST-GCN, no node gate	2	−0.0766	6.435 $\times 10^{-79}$	−12.156/5.350 $\times 10^{-34}$
ST-GCN, observed graph–ST-GCN, no node gate	3	−0.0283	2.985 $\times 10^{-5}$	−4.719/2.366 $\times 10^{-6}$

Table 10 confirms that the final XGBoost objective improves the tabular baseline relative to the earlier squared-error results but does not close the gap to the neural models under paired forecast-window testing. The positive ST-GCN–LSTM MAE differences show that LSTM remains better for absolute-error concentration, whereas Tables 5 and 6 show that graph-aware variants remain useful for dynamic sensitivity and propagation analysis.

The node-gating ablation provides an empirical check on whether airport-specific output modulation contributes to performance. If the gated model improves over the ungated variant, this supports the interpretation that heterogeneous airport sensitivity is relevant for delay propagation forecasting. If the difference is small, the node gate should be interpreted mainly as a lightweight calibration layer rather than as a major architectural contribution.

Table 10. Additional paired tests for the final run using window-level MAE loss.

Comparison (A–B)	h	Δ MAE	Wilcoxon p	DM-Style p
ST-GCN, observed graph–XGBoost, spatial features	1	−3.3764	<0.001	<0.001
ST-GCN, observed graph–XGBoost, spatial features	2	−3.6500	<0.001	<0.001
ST-GCN, observed graph–XGBoost, spatial features	3	−3.6522	<0.001	<0.001
LSTM–XGBoost, spatial features	1	−3.6720	<0.001	<0.001
LSTM–XGBoost, spatial features	2	−3.6756	<0.001	<0.001
LSTM–XGBoost, spatial features	3	−3.6686	<0.001	<0.001
ST-GCN, observed graph–LSTM	1	0.2957	<0.001	<0.001
ST-GCN, observed graph–LSTM	2	0.0256	<0.001	<0.001
ST-GCN, observed graph–LSTM	3	0.0164	<0.001	<0.001

3.2. Regime-Specific Performance

Operational decision-making is driven by conditional performance across delay magnitudes, particularly in the disruption tail. Table 11 reports error stratification for $h = 1$ over bins of true delay in the following minutes: 0–5, 5–15, 15–30, 30–60, and 60–180. The error structure is strongly heteroscedastic, mainly because all models degrade sharply as delays move into the 60–180 min regime, reflecting a mixture of exogenous shocks, cascading constraints, and saturation under clipping and log transforms.

Table 11. Regime-specific performance for $h = 1$ by true-delay bin.

Model	Bin (min)	Count	RMSE (min)	MAE (min)	Underprediction Rate	Horizon
ST-GCN, observed graph	0–5	1,766,678	4.5187	0.5661	0.0090	1
ST-GCN, observed graph	5–15	32,431	16.4560	10.9482	0.6648	1
ST-GCN, observed graph	15–30	23,059	21.7532	17.8421	0.7712	1
ST-GCN, observed graph	30–60	18,556	36.3609	32.4696	0.8772	1
ST-GCN, observed graph	60–180	15,936	102.0147	90.3303	0.9595	1
LSTM	0–5	1,766,678	1.0531	0.1168	0.0126	1
LSTM	5–15	32,431	8.9682	8.1631	0.8758	1
LSTM	15–30	23,059	19.6223	18.4634	0.9631	1
LSTM	30–60	18,556	40.2776	38.8105	1.0000	1
LSTM	60–180	15,936	111.0251	103.0796	1.0000	1
XGBoost, spatial features	0–5	1,766,678	14.7506	4.2214	0.0040	1
XGBoost, spatial features	5–15	32,431	14.0913	8.7397	0.5500	1
XGBoost, spatial features	15–30	23,059	17.4230	14.2011	0.8100	1
XGBoost, spatial features	30–60	18,556	33.4882	30.3932	0.9419	1
XGBoost, spatial features	60–180	15,936	100.9004	90.6518	0.9879	1

Table 11 shows two regime-level signatures. First, LSTM is strongest in the 0–5 min near-on-time mass, with RMSE 1.0531 min and MAE 0.1168 min, but it underpredicts every 60–180 min case in this bin-specific evaluation. Second, all models deteriorate sharply in the severe-delay tail. XGBoost with spatial features has the lowest 60–180 min RMSE among the three models shown, while the observed-graph ST-GCN has a slightly lower 60–180 min MAE; however, both models retain high underprediction rates, so the tail improvement does not remove the operational risk of missed severe escalation.

Table 12 identifies where the static ST-GCN fails most. The dominant failure mode is severe-delay compression: for true delays of at least 60 min, the mean prediction is only 17.35 min, producing a bias of -87.83 min and an underprediction rate of 0.9595. Weather-affected periods, high-connectivity airports, long-distance routes, and high-traffic airports have substantially larger errors than their low-traffic or short-route counterparts. The severe-underprediction columns further show that many true severe events remain below 30 or 60 min in the forecast, particularly outside hubs and in lower-connectivity strata.

Table 12. Severe-delay context analysis for ST-GCN with the observed graph at $h = 1$.

Context	Stratum	Count	True Mean	Pred. Mean	Bias	MAE	RMSE	Underprediction Rate	Pred. <30	Pred. <60
Threshold	≥ 30	34,492	71.17	16.51	−54.66	59.20	74.29	0.9152	0.7854	0.9122
Threshold	≥ 60	15,936	105.18	17.35	−87.83	90.33	102.01	0.9595	0.7854	0.9122
Hub status	hub	185,666	10.40	11.39	0.99	12.91	25.54	0.2563	0.6088	0.8400
Hub status	non-hub	1,670,994	0.82	0.04	−0.78	0.84	8.65	0.0235	0.9941	0.9975
Connectivity	high	617,214	4.89	3.54	−1.34	5.70	18.70	0.1299	0.7612	0.9023
Connectivity	mid	617,214	0.40	0.00	−0.39	0.40	6.40	0.0093	1.0000	1.0000
Connectivity	low	622,232	0.07	0.00	−0.07	0.07	2.85	0.0014	1.0000	1.0000
Time of day	morning peak	358,530	1.32	0.40	−0.92	1.39	9.39	0.0515	0.9718	0.9970
Time of day	midday	539,460	2.33	1.41	−0.92	2.66	13.01	0.0619	0.8462	0.9472
Time of day	evening peak	449,550	2.67	2.64	−0.03	3.28	14.32	0.0495	0.6184	0.8295
Time of day	night	509,120	0.73	0.19	−0.55	0.78	7.81	0.0249	0.9349	0.9792
Weather	weather	4708	25.89	22.23	−3.65	23.29	40.92	0.3148	0.3870	0.6807
Weather	non-weather	1,851,952	1.72	1.12	−0.59	2.00	11.34	0.0461	0.8060	0.9242
Route distance	long	617,214	4.33	3.45	−0.88	5.14	17.38	0.1150	0.7328	0.8887
Route distance	medium	617,214	0.89	0.09	−0.79	0.89	9.01	0.0226	0.9735	0.9976
Route distance	short	622,232	0.14	0.00	−0.14	0.14	3.95	0.0030	1.0000	1.0000
Traffic class	high	617,214	4.88	3.54	−1.34	5.70	18.68	0.1299	0.7609	0.9022
Traffic class	mid	617,214	0.39	0.00	−0.39	0.39	6.35	0.0092	1.0000	1.0000
Traffic class	low	622,232	0.08	0.00	−0.08	0.08	3.07	0.0016	1.0000	1.0000

Table 13 shows that the hybrid combination improves severe-delay MAE relative to the standalone neural models at $h = 1$, but it does not solve severe-delay underprediction. The hybrid model gives the lowest severe-delay MAE among the rows shown and reduces

the fraction of true ≥ 60 min cases predicted below 30 min relative to the standalone models, yet the remaining underprediction rates remain operationally high.

Table 13. Additional severe-delay metrics for $h = 1$ after the validation-selected optimization run.

Model	Count	MAE, ≥ 60 min	RMSE, ≥ 60 min	Underprediction Rate	Pred. < 30	Pred. < 60
ST-GCN, observed graph	15,936	90.33	102.01	0.960	0.785	0.912
LSTM	15,936	103.08	111.03	1.000	1.000	1.000
XGBoost, spatial features	15,936	90.65	100.90	0.988	0.845	0.962
Hybrid ST-GCN + XGBoost	15,936	88.35	99.54	0.980	0.778	0.943

3.3. Qualitative Error Analysis

Error analysis focuses on residual distributions, calculated by subtracting the true delay from the predicted delay. Figures 4–6 use density-oriented histograms on the original minute scale, with internal chart titles removed so that the manuscript captions carry the interpretation. The LSTM histogram given in Figure 4 has a sharp spike near zero, giving the high accuracy for small delays, and a pronounced negative tail, consistent with the systematic underprediction of large delays.

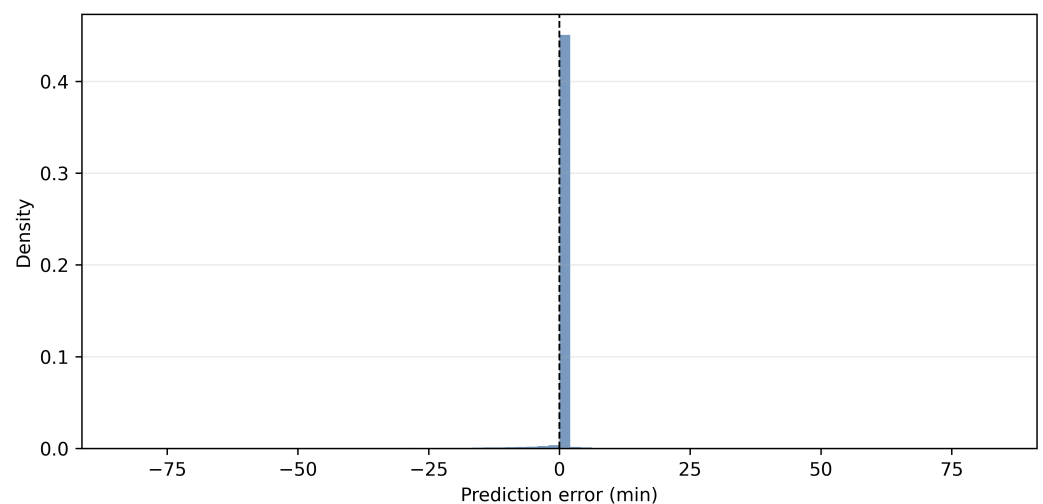


Figure 4. LSTM error histogram for $h = 1$ in minutes.

The ST-GCN histogram in Figure 5 is wider with extended tails, consistent with higher variance predictions and occasional saturation under the ceiling cap.

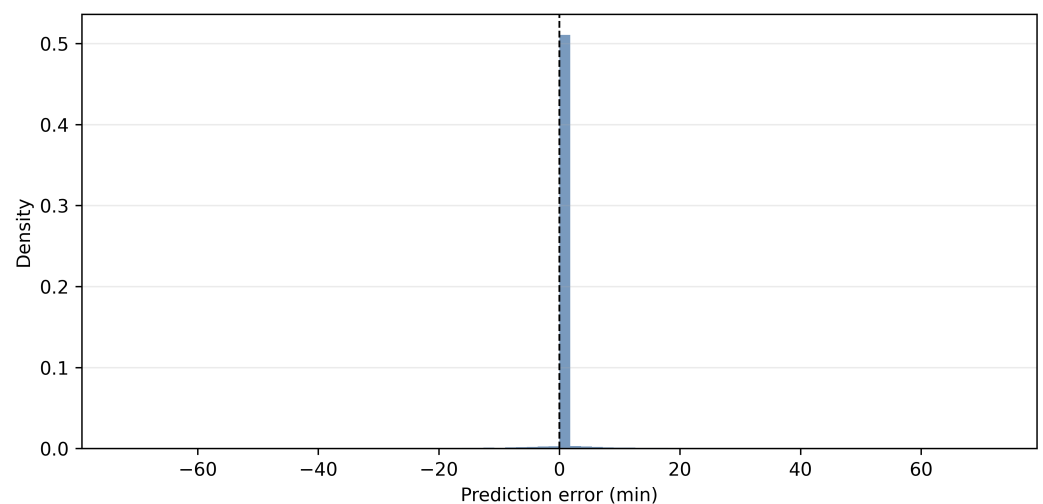


Figure 5. ST-GCN error histogram for $h = 1$ in minutes.

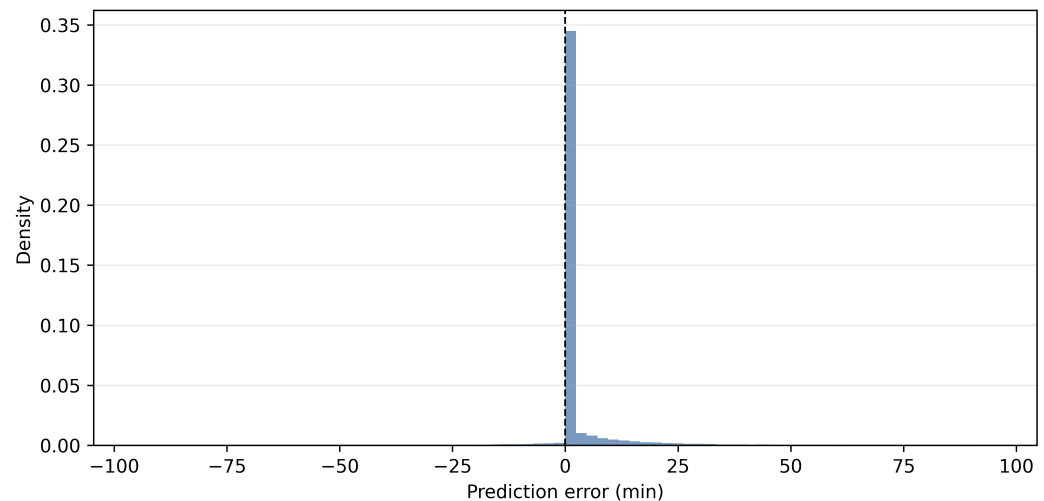


Figure 6. XGBoost error histogram for $h = 1$ in minutes.

The XGBoost histogram in Figure 6 retains a broad residual dispersion under the aligned final run. In combination with Table 11, this indicates that the direct XGBoost model can reduce some severe-bin RMSE relative to the neural models while still producing weaker overall point accuracy and very high severe-delay underprediction.

Rare-event detection is evaluated under severe imbalance, with overall prevalence 0.008722 and higher prevalence in the validation and test sets, as reported in Table 1. Under the temperature scaling given by Equation (23) with $T^* = 1.200$, the final one-hour ST-GCN classifier has test ROC-AUC 0.8856, PR-AUC 0.1292, and Brier score 0.0127. Because the positive class is rare, PR-AUC, precision, recall, and threshold sensitivity are reported in addition to ROC-AUC.

Threshold selection by validation F_1 yields $\tau^* = 0.125$ for the one-hour ST-GCN classifier. At τ^* , test precision is 0.1523, recall is 0.3028, and F_1 is 0.2026. This operating point reflects an engineering trade in which recall is prioritized relative to precision under a low base rate, while still avoiding the very high alert rates produced by very low thresholds. Under operational deployment, such a detector is most appropriate as a triage component coupled to downstream decision rules that incorporate capacity, cost asymmetry, and human-in-the-loop verification.

Table 14 shows that ranking performance can remain reasonable even when precision is low, which is expected under severe class imbalance. The one-hour ST-GCN classifier has a higher PR-AUC than the dynamic-hourly classifier at the selected operating point, while regression-derived detection of delays ≥ 60 min remains weak for both ST-GCN and XGBoost. These results support treating severe-disruption alerting as a threshold-dependent operational decision rather than as a simple binary accuracy problem.

Table 14. Rare-event and severe-delay detection metrics on the test split for $h = 1$.

Model	Event Score	Threshold	ROC-AUC	PR-AUC	Precision	Recall	F_1
ST-GCN, observed graph	Rare-event classifier	0.125	0.8856	0.1292	0.1523	0.3028	0.2026
ST-GCN, hourly dynamic graph	Rare-event classifier	0.090	0.8767	0.0884	0.1169	0.3905	0.1800
ST-GCN, observed graph	Predicted delay ≥ 30 min	30.000	0.7708	0.1732	0.3082	0.2067	0.2475
ST-GCN, observed graph	Predicted delay ≥ 60 min	60.000	0.7412	0.0850	0.2392	0.0897	0.1305
XGBoost, spatial features	Predicted delay ≥ 30 min	30.000	0.7630	0.0644	0.0680	0.1347	0.0904
XGBoost, spatial features	Predicted delay ≥ 60 min	60.000	0.7411	0.0293	0.0235	0.0377	0.0289

Figure 7 complements Table 14 by showing how precision, recall, and F_1 change with the operating threshold. The curves make explicit that no single threshold is universally optimal: lower thresholds increase recall at the cost of false alerts, whereas higher thresholds improve precision but miss more severe events.

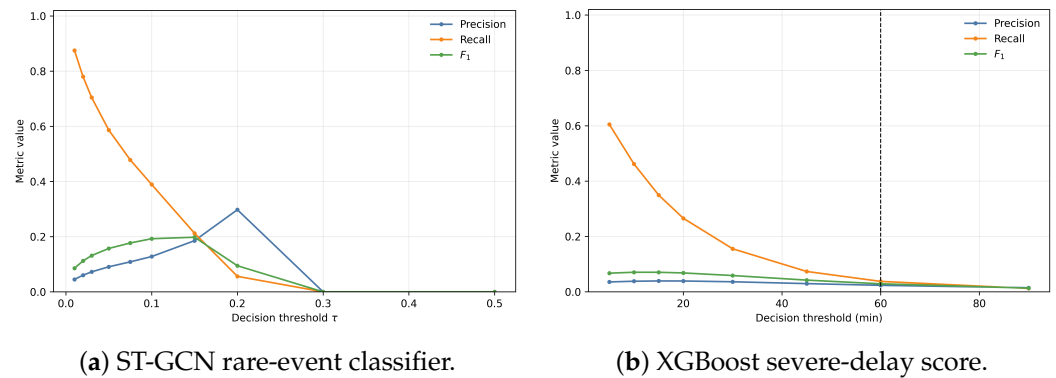


Figure 7. Threshold sensitivity for rare-disruption and severe-delay detection at $h = 1$.

Conformal prediction intervals defined by Equation (22) provide marginal finite-sample coverage under exchangeability assumptions between calibration and deployment distributions. The reported absolute error quantiles are 0.00 min at 90% and 6.72 min at 95%, with empirical test coverages approximately 89.8% and 94.6%. The near-zero 90% quantile indicates that a large portion of evaluated instances incur negligible absolute error, consistent with the near-on-time flights dominating the data-generating process, while the 95% quantile provides a practical risk bound for moderate deviations. In an air traffic management context, these intervals can be interpreted as conservative uncertainty envelopes on predicted departure delay, suitable for risk-aware scheduling and contingency planning where false confidence can induce costly downstream propagation.

3.4. Discussion

The reported $h = 1$ errors from ST-GCN with the observed graph (RMSE 11.5141 min, MAE 2.0513 min), LSTM (RMSE 11.3691 min, MAE 1.7556 min), XGBoost with temporal features (RMSE 17.6896 min, MAE 5.4273 min), and XGBoost with spatial features (RMSE 17.6881 min, MAE 5.4277 min) indicate that the aligned final evaluation is dominated by the neural temporal learners rather than by the direct tabular baselines. The graph model provides competitive multi-horizon RMSE values at a substantially lower parameter count than the LSTM, while LSTM remains strongest in MAE.

The comparative results should not be interpreted as evidence that the ST-GCN uniformly dominates all baselines. The revised experiments show a more nuanced pattern. In the aligned final evaluation, LSTM is strongest in point MAE and RMSE among the primary forecasting models, while the direct XGBoost baselines provide nonlinear tabular benchmarks but do not dominate the neural models. LSTM remains competitive in MAE because the dominant mass of the target distribution is close to zero, making temporal smoothing effective for common near-on-time regimes. The advantage of ST-GCN is therefore not universal point-forecast superiority but network-aware representation: it explicitly propagates information across the airport graph, supports multi-horizon prediction within a single model, provides lower parameter complexity than the recurrent baseline, and produces node-level outputs that can be linked to disruption monitoring and network resilience analysis.

To assess whether the observed performance differences are statistically meaningful, the analysis adds bootstrap confidence intervals and paired forecast-window comparisons. Bootstrap intervals are computed by resampling forecast windows rather than individual airport-node observations, preserving the dependence structure within each forecast window. Paired Wilcoxon tests and Diebold–Mariano-style comparisons are then applied to matched window-level loss differentials between ST-GCN, LSTM, and XGBoost. These tests are essential because small differences in RMSE or MAE can appear numerically

meaningful when millions of node-time observations are flattened, even when the practical difference between models is weak.

The relatively stable error across the 1–3 h horizons can be explained by the aggregation and target formulation. First, hourly airport-level aggregation smooths flight-level idiosyncrasies and converts many individual schedule deviations into a slower-moving airport-state signal. Second, the train-only calendar baseline characterizes predictable daily, weekly, and monthly structure for disruption labeling and residual tabular modeling, while the neural regressors forecast the future departure delay signal in log-transformed minutes. Third, a three-hour horizon remains within the short tactical range where delay persistence, airport load, and recent upstream delay remain informative. This stability should not be misread as evidence that long-range forecasting is solved; rather, it indicates that the selected horizons remain close enough to the current operational state for persistence and network structure to remain useful.

From an operational perspective, errors of approximately 10–12 min are potentially useful for network-level monitoring, flow-management awareness, and disruption triage, but they are not precise enough for deterministic flight-level gate assignment, aircraft-turnaround control, or crew decisions. The framework is best understood as an early-warning and prioritization tool. It can identify airports or time windows in which delay risk is increasing and where human operators may need to inspect capacity, weather, aircraft rotations, or flow restrictions. Thus, the practical value of the model lies less in exact minute-level prediction for every airport and more in supporting tactical situational awareness across a large national network.

The severe-delay regimes reveal the main limitation of all evaluated models. Prediction performance deteriorates for 60–180 min delays because these events are rare, heterogeneous, and often driven by exogenous or partially unobserved mechanisms such as convective weather, airspace restrictions, runway configuration changes, airport acceptance rate reductions, crew and aircraft rotation failures, and cascading network recovery constraints. Log-space regression and clipping improve numerical stability but can also compress the largest events, increasing the risk of underprediction in the tail. LSTM tends to smooth toward the dominant near-zero regime, which explains its strong MAE but weak severe-delay behavior. XGBoost with spatial features has a lower 60–180 min RMSE than the neural models in Table 11, but it still severely underpredicts the tail. ST-GCN provides a more structured network representation, but graph smoothing can dilute localized extreme disruptions unless the dynamic graph and external operational variables capture the disruption mechanism sufficiently.

The rare-disruption classifier must also be interpreted under severe class imbalance. Table 14 and Figure 7 show that a high ROC-AUC can coexist with modest PR-AUC and low precision because the severe-event base rate is small. In operational use, this may still be acceptable if the model is used as a triage layer where missed severe disruptions are more costly than false alarms. However, it would not be acceptable as an automatic intervention trigger without additional cost-sensitive decision rules and threshold selection reflecting local operational costs.

The additional final results in Tables 5, 10 and 13 reinforce this interpretation. The XGBoost objective improved the tabular baseline relative to the earlier squared-error run, and the hybrid combination reduced severe-delay MAE at $h = 1$, but neither change eliminated the high underprediction rate for true ≥ 60 min delays. This suggests that the main scientific limitation is not only the choice of loss function but the lack of high-resolution exogenous operational inputs that explain rare severe disruptions.

Comparison with Previous Studies

Across comparable network-wide delay studies, headline error values are not directly comparable without harmonizing target definitions, aggregation intervals, spatial coverage, and evaluation splits. Baseline subtraction, non-negativity clipping, and log transforms can reduce typical absolute errors while leaving tail-risk prediction difficult. Nevertheless, a comparative analysis is useful for assessing the scale and relevance of the present results. Table 15 situates the present study against representative network-wide delay forecasting works that use hourly or sub-hourly aggregation, smaller airport subsets ($N \approx 70$ – 75), and targets closer to raw averaged delay rather than train-only calendar-baseline excess delay with $\log(1 + y)$ regression. The present multi-horizon setting uses $t_{in} = 6$ and $t_{out} = 3$ (1–3 h ahead) and evaluates spatio-temporal graph models, recurrent baselines, and boosted-tree tabular baselines.

Table 15. Dataset and experimental setup comparison across representative works.

Field	Present Study	[32] STPN	[31] AG2S-Net	[33] STGMAGNet	[34] CausalNet
Years	2019–2023	2015–2021	5 y	2019	2018
N° Flights	2.88 M flights	Not Reported	32,868 states	Not Reported	2.19 M flights
Bin	60 m	30 m	1 h	1 h	1 h
T	35,261	Not Reported	32,868	Not Reported	5136
N	370	70	75	75	74
E	5334	Not Reported	242	Not Reported	Not Reported
Target	excess, clip ≥ 0 , log1p	avg delay, clip ≤ 30	link avg delay	avg delay (early = 0)	avg delay
Horizons	1–3 h (6→3) time:	1.5/3/6 h (12→12)	Not Reported	24 h (24→24)	1–3 h
Split	[0,22,671]/[22,671,30,235]/[30,235,35,261)	70/10/20 chrono	Not Reported	Not Reported	70/15/15

Table 16 reports the closest available point-forecast metrics from the present study and from representative graph-based delay-prediction studies. The comparison should be interpreted as contextual rather than as a strict leaderboard, since the datasets, targets, horizons, clipping rules, and train–test partitions differ across studies. Even with these differences, the table shows that the present neural models achieve competitive short-horizon MAE values while operating on a substantially larger airport network. The final aligned evaluation places LSTM ahead in point-error metrics, whereas ST-GCN provides graph-aware multi-horizon performance with much lower parameter complexity than the LSTM baseline.

Table 16. Performance comparison with representative flight delay forecasting studies.

Study/Context	Model	Horizon	RMSE (min)	MAE (min)	Notes	Reference
Present study	LSTM	1 h	11.37	1.76	Flattened $N \times F$ input; 1.23 M parameters	Present study
Present study	ST-GCN	1 h	11.51	2.05	Graph convolution; 70,058 parameters	Present study
Present study	XGBoost temporal	1 h	17.69	5.43	Direct temporal-only tabular model; $N = 370$	Present study
Present study	XGBoost spatial	1 h	17.69	5.43	Direct spatial-feature tabular model; $N = 370$	Present study
Network-wide delay prediction	STPN	1.5 h	–	7.20	Average delay clipped to 30 min; $N = 70$	[32]
Network-wide delay prediction	ST-GCN	1.5 h	–	7.78	Same dataset and target definition as STPN comparison	[32]
Causal spatio-temporal prediction	CausalNet	1 h	–	5.62	China airport network; average delay target; $N = 74$	[34]
Causal spatio-temporal prediction	ST-GCN	1 h	–	6.50	Baseline graph model in the same experimental setting	[34]

The final aligned comparison does not support the claim that XGBoost outperforms learned spatio-temporal models in point-error metrics. Instead, the XGBoost variants serve as direct nonlinear tabular stress tests showing that temporal and spatial engineered features alone are insufficient to match the neural temporal learners under the present window alignment and target definition.

The empirical results demonstrate that the proposed framework achieves competitive predictive accuracy while operating at substantially larger spatial and temporal scales

than most prior studies. Nevertheless, the results should be interpreted cautiously. LSTM achieves the lowest aligned point-error metrics, and the XGBoost direct baselines do not dominate the neural models. The advantage of ST-GCN is therefore not universal point-forecast superiority. Its value lies in graph-aware multi-horizon prediction, lower parameter complexity relative to LSTM, explicit representation of airport connectivity, and integration with disruption detection and uncertainty quantification. The small differences among neural models require statistical qualification; therefore, confidence intervals and paired error tests over forecast windows were added.

The model is trained on a dataset comprising 2,878,854 flights across 370 airports and 5334 directed route connections, covering a multi-year period from 2019 to 2023. This scale substantially exceeds that of many previously reported airport-network delay forecasting experiments, which are often restricted to subsets of major hubs or shorter observation windows. Despite this expanded spatial coverage and the pronounced variability inherent to network-level operations, the static ST-GCN achieves a test mean absolute error of 2.0513 min at the one-hour horizon, while the dynamic-hourly sensitivity variant reduces one-hour RMSE to 11.4086 min. The results therefore demonstrate that accurate short-horizon forecasting and graph-structure sensitivity analysis are achievable even under large-scale, heterogeneous network conditions.

An additional strength of the framework lies in its representational efficiency. The selected spatio-temporal graph convolutional configuration contains 70,058 trainable parameters, which remains much smaller than the recurrent baseline with approximately 1.23 M parameters, yet produces competitive predictive accuracy. This result suggests that explicitly encoding the airport connectivity structure through graph operators provides a parameter-efficient mechanism for representing delay propagation compared with relying solely on high-capacity recurrent representations. The findings therefore reinforce the importance of network-aware learning architectures in modeling transportation systems where interactions between nodes constitute a primary driver of system dynamics.

From an operational perspective, the scale and resolution of the dataset used in this study provide a particularly strong empirical basis for evaluating forecasting models. The dataset spans nearly three million flights and includes detailed operational attributes such as departure delays, arrival delays, taxi times, and weather-related delay components derived from the U.S. Department of Transportation reporting system. Such breadth enables the model to learn both local airport dynamics and network-level propagation patterns across diverse operational regimes. Consequently, the results provide evidence that graph-based learning approaches can deliver reliable short-term delay forecasts across a national-scale air transportation network while maintaining computational efficiency and statistical robustness. An error of approximately 10–12 min may be operationally acceptable for strategic and tactical monitoring of airport-level disruption risk, but it is not sufficient for precise gate-level or aircraft-turnaround control. The framework is therefore best interpreted as a network-level early-warning tool rather than a deterministic flight-release decision system.

4. Conclusions

This study investigated short-horizon airport-level delay propagation across the U.S. domestic air transportation network using a large-scale spatio-temporal machine learning framework. The analysis used 2,878,854 flight records from January 2019 to August 2023 and constructed an hourly airport-network representation with 370 airport nodes and 5334 directed route connections. The regression task was formulated as multi-horizon forecasting of future airport-level departure delay, while a train-only calendar baseline was used to define excess-delay disruption labels and regime analyses. This formulation

allowed the evaluation to retain operational minute-scale departure delay forecasts while separately identifying deviations from routine schedule-driven patterns.

The results show that no single model dominates all the evaluation criteria. In the aligned final evaluation, LSTM provides the lowest point-error metrics among the primary models, while the direct XGBoost temporal-only and spatial-feature baselines do not dominate the neural models. The ST-GCN does not consistently outperform every baseline in point-forecast accuracy, and the manuscript therefore avoids claiming universal superiority. Its principal contribution is different: it provides a graph-aware, parameter-efficient, multi-horizon representation of airport delay propagation that explicitly encodes connectivity among airports and supports network-level interpretation, graph sensitivity analysis, and node-level disruption monitoring.

The ablation analysis strengthens this interpretation. Comparing the observed route-count adjacency with identity/no-graph, randomized-adjacency, dynamic/hour-conditioned adjacency, no-graph-feature, no-node-gate, and spatial XGBoost variants allows the contribution of the graph structure to be separated from temporal persistence and tabular nonlinear learning. The dynamic-hourly sensitivity variant improves RMSE relative to the static ST-GCN, whereas MAE differences remain mixed because near-zero delay states dominate the target distribution. In this sense, graph neural networks provide information that traditional tabular methods do not naturally expose: they represent how airport states interact through route connectivity, how upstream and downstream delay pressure can be propagated, and which parts of the network may become operationally critical during disruption.

Beyond point forecasting, the framework integrates rare-disruption detection and uncertainty quantification. The disruption-detection component provides a threshold-dependent early-warning layer for identifying airport-time states with elevated risk, and the revised evaluation reports ROC-AUC, PR-AUC, precision, recall, F_1 , and threshold sensitivity to reflect the rare-event base rate. Conformal prediction intervals provide calibrated uncertainty estimates that can support risk-aware operational planning. These uncertainty estimates are particularly important because point forecasts alone can create false confidence in rare severe-delay regimes, where errors are larger and operational consequences are more significant.

The validation-selected optimization run supports the same central conclusion while refining the quantitative interpretation. The neural objectives remain the most defensible choices under the validation composite score, while XGBoost improves the tabular baselines but does not overtake the neural models in global point accuracy. Severe-delay MAE improves most in the hybrid ST-GCN–XGBoost result at $h = 1$, but the remaining under-prediction rates show that severe disruption forecasting still requires richer operational covariates rather than loss-function tuning alone.

The framework has several practical applications. For airport operation management, it can support monitoring of emerging delay pressure and prioritization of airports requiring closer inspection. For air traffic flow control, it can provide short-horizon network-level indicators of where delay propagation may intensify. For airline and airport resource allocation, it can inform gate, ground-handling, passenger-reaccommodation, and recovery-planning decisions. For disruption management, it can help identify whether a delay pattern is isolated or likely to propagate across connected airports. For network resilience assessment, it provides a data-driven representation of how operational stress moves through a national air transportation system.

Several limitations remain. First, although the analysis includes static, randomized, no-graph, and dynamic/hour-conditioned graph variants, the main architecture still depends on historical operational data and does not fully model causal intervention mechanisms.

Second, the forecasting horizon is limited to the short tactical range of 1–3 h. Third, severe delays remain difficult because they are rare, heterogeneous, and often driven by exogenous mechanisms not fully represented in the available dataset. Fourth, conformal prediction intervals rely on calibration assumptions that may be weakened by temporal distribution shift. Fifth, the available features do not fully represent all relevant meteorological, airspace, runway, and traffic-management constraints.

Future extensions should fuse high-resolution meteorological fields, convective-weather nowcasts, airport acceptance rates, runway configuration, airspace flow restrictions, sector congestion, and traffic-management initiatives. The framework is transferable to other countries if equivalent flight schedules, airport metadata, and route graphs are available, but performance may change in networks with different hub structures, slot constraints, weather regimes, and data completeness. Future work should also investigate longer horizons, explicitly cost-sensitive disruption detection, dynamic graph-learning mechanisms, distributional forecasting, and causal evaluation of delay propagation interventions. Overall, this study demonstrates that graph-based spatio-temporal learning is a valuable component of large-scale aviation delay forecasting, especially when the goal is not only to minimize average error but also to understand, monitor, and manage disruption propagation across an interconnected airport network.

Author Contributions: Conceptualization, L.F.F.M.S. and S.G.; methodology, L.F.F.M.S.; software, L.F.F.M.S.; validation, L.F.F.M.S. and S.G.; formal analysis, L.F.F.M.S.; investigation, L.F.F.M.S.; resources, L.F.F.M.S.; data curation, L.F.F.M.S.; writing—original draft preparation, L.F.F.M.S.; writing—review and editing, L.F.F.M.S. and S.G.; visualization, L.F.F.M.S.; supervision, S.G.; project administration, L.F.F.M.S.; funding acquisition, L.F.F.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundação para a Ciência e a Tecnologia through projects 10.54499/UIDB/50022/2020, 10.54499/UIDP/50022/2020, and 10.54499/LA/P/0079/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The flight delay dataset analyzed in this study is publicly available as the Flight Delay and Cancellation Dataset (2019–2023) on Kaggle [40].

Acknowledgments: The present work was performed under the scope of activities at the Aeronautics and Astronautics Research Center (AEROG) and Institute of Mechanical Engineering (IDMEC) of the Laboratório Associado em Energia, Transportes e Aeroespacial (LAETA).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bureau of Transportation Statistics. Air Travel Consumer Report: December 2024, Full Year 2024 Numbers. 2025. Available online: <https://www.bts.gov/newsroom/air-travel-consumer-report-december-2024-full-year-2024-numbers> (accessed on 1 June 2026).
2. Bureau of Transportation Statistics. Airline On-Time Performance and Causes of Flight Delays. 2021. Available online: <https://www.bts.gov/explore-topics-and-geography/topics/airline-time-performance-and-causes-flight-delays> (accessed on 1 June 2026).
3. Bureau of Transportation Statistics. Technical Directive Number 15: Gate Departure Time. 2024. Available online: <https://www.bts.gov/topics/airlines-and-airports/number-15-gate-departure-time> (accessed on 1 June 2026).
4. Federal Aviation Administration. Air Traffic by the Numbers. 2022. Available online: https://www.faa.gov/sites/faa.gov/files/air_traffic/by_the_numbers/Air_Traffic_by_the_Numbers_2022.pdf (accessed on 1 June 2026).
5. Federal Aviation Administration. Air Traffic by the Numbers: Fiscal Year 2024. 2025. Available online: https://www.faa.gov/air_traffic/by_the_numbers/air-traffic-by-the-numbers-FY2024.pdf (accessed on 1 June 2026).
6. Fleurquin, P.; Ramasco, J.J.; Eguiluz, V.M. Systemic Delay Propagation in the US Airport Network. *Sci. Rep.* **2013**, *3*, 1159. [CrossRef] [PubMed]

7. Pyrgiotis, N.; Malone, K.M.; Odoni, A. Modelling Delay Propagation within an Airport Network. *Transp. Res. Part C Emerg. Technol.* **2013**, *27*, 60–75. [\[CrossRef\]](#)
8. Kafle, N.; Zou, B. Modeling Flight Delay Propagation: A New Analytical-Econometric Approach. *Transp. Res. Part B Methodol.* **2016**, *93*, 520–542. [\[CrossRef\]](#)
9. Shone, R.; Glazebrook, K.; Zografos, K.G. Applications of stochastic modeling in air traffic management: Methods, challenges and opportunities for solving air traffic problems under uncertainty. *Eur. J. Oper. Res.* **2021**, *292*, 1–26. [\[CrossRef\]](#)
10. Zhang, Y.; Lee, S.Y.; Gu, Y. A review of air transport service quality studies: Current status and future research agenda. *J. Air Transp. Res. Soc.* **2023**, *1*, 9–21. [\[CrossRef\]](#)
11. Milioti, C.; Odoni, A.R. Airport “level” and market concentration in Europe. *J. Air Transp. Res. Soc.* **2024**, *2*, 100007. [\[CrossRef\]](#)
12. Beatty, R.; Hsu, R.; Berry, L.; Rome, J. Preliminary Evaluation of Flight Delay Propagation through an Airline Schedule. *Air Traffic Control Q.* **1999**, *7*, 259–270. [\[CrossRef\]](#)
13. Wang; Schaefer; Wojcik. Flight connections and their impacts on delay propagation. In *Proceedings of the 22nd Digital Avionics Systems Conference Proceedings (Cat No 03CH37449) DASC-03*; IEEE: New York, NY, USA, 2003; pp. 5.B.4–5.1-9. [\[CrossRef\]](#)
14. Brueckner, J.K.; Czerny, A.I.; Gaggero, A.A. Airline delay propagation: A simple method for measuring its extent and determinants. *Transp. Res. Part B Methodol.* **2022**, *162*, 55–71. [\[CrossRef\]](#)
15. Giannikas, V.; Ledwoch, A.; Stojković, G.; Costas, P.; Brintrup, A.; Al-Ali, A.A.S.; Chauhan, V.K.; McFarlane, D. A data-driven method to assess the causes and impact of delay propagation in air transportation systems. *Transp. Res. Part C Emerg. Technol.* **2022**, *143*, 103862. [\[CrossRef\]](#)
16. Chen, G.; Fricke, H.; Okhrin, O.; Rosenow, J. Flight delay propagation inference in air transport networks using the multilayer perceptron. *J. Air Transp. Manag.* **2024**, *114*, 102510. [\[CrossRef\]](#)
17. Dias, N.G.; Santos, L.F.F.M.; Melicio, R. Aircraft Maintenance Professionals: Stress, Pressure and Fatigue. *Proc. MATEC Web Conf.* **2019**, *304*, 06001. [\[CrossRef\]](#)
18. da Silva, J.C.; Santos, L.F.F.M.; Sousa, J.M.C.; Silva, J. Human Fatigue in the Aircraft Maintenance Environment: A Systematic Literature Review. *Saf. Sci.* **2024**, *180*, 106634. [\[CrossRef\]](#)
19. Lázaro, F.L.; Madeira, T.; Melicio, R.; Valério, D.; Santos, L.F.F.M. Identifying Human Factors in Aviation Accidents with Natural Language Processing and Machine Learning Models. *Aerospace* **2025**, *12*, 106. [\[CrossRef\]](#)
20. Wang, X.; Brownlee, A.E.; Woodward, J.R.; Weiszer, M.; Mahfouf, M.; Chen, J. Aircraft taxi time prediction: Feature importance and their implications. *Transp. Res. Part C Emerg. Technol.* **2021**, *124*, 102892. [\[CrossRef\]](#)
21. Yin, J.; Zhang, M.; Ma, Y.; Wu, W.; Li, H.; Chen, P. Prediction and Analysis of Airport Surface Taxi Time: Classification, Features, and Methodology. *Appl. Sci.* **2024**, *14*, 1306. [\[CrossRef\]](#)
22. Cai, Q.; Alam, S.; Duong, V.N. A Spatial-Temporal Network Perspective for the Propagation Dynamics of Air Traffic Delays. *Engineering* **2021**, *7*, 452–464. [\[CrossRef\]](#)
23. Lázaro, R.R.; Santos, L.F.F.M. Fine-Tuning BERT for Aviation Accident Report Classification: A Comparative Study of AI-Driven Analysis Techniques. *Appl. Sci.* **2025**, *15*, 9403. [\[CrossRef\]](#)
24. Samarra, F.; Santos, L.F.F.M.; Melicio, R. Correlations between Socioeconomic Indicators and Aviation Accidents. *Appl. Sci.* **2023**, *13*, 7997. [\[CrossRef\]](#)
25. Santos, L.F.F.M.; Costa, J.A.; Valério, D.; Batista, N.; Melicio, R. Blockchain Information Based Systems in Aviation: The Advantages for Aircraft Records Management. In *Proceedings of the 2022 International Conference on Control, Automation and Diagnosis (ICCAD)*; IEEE: New York, NY, USA, 2022; pp. 1–6. [\[CrossRef\]](#)
26. Song, Y.; Hu, X. Learning-Based Demand-Supply-Coupled Charging Station Location Problem for Electric Vehicle Demand Management. *Transp. Res. Part D. Transp. Environ.* **2023**, *125*, 103975. [\[CrossRef\]](#)
27. Wang, J.; Kaushik, H.D.; Jacob, R.A.; Zhang, J. Spatiotemporal Planning of Electric Vehicle Charging Infrastructure: Demand Estimation and Grid-Aware Optimization under Uncertainty. *iScience* **2025**, *28*, 113368. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Carbonera, M.; Ciavotta, M.; Messina, E. Variational Autoencoders and Generative Adversarial Networks for Multivariate Scenario Generation. *Data Sci. Transp.* **2024**, *6*, 23. [\[CrossRef\]](#)
29. Carbonera, M.; Ciavotta, M.; Messina, E. Generative AI for Realistic Urban Traffic Scenario Generation: β -GCN-VAE and β -Transformer-VAE Models. **2026**, *Research Square preprint*. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Shangguan, A.; Feng, N.; Hei, X.; Fei, R.; Jin, Y.; Mu, L.; Li, Y. Predicting Road Traffic Accident Severity from Imbalanced Data Using VAE Attention and GCN. *Sci. Rep.* **2025**, *15*, 34372. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Bao, J.; Yang, Z.; Zeng, W. Graph to sequence learning with attention mechanism for network-wide multi-step-ahead flight delay prediction. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103323. [\[CrossRef\]](#)
32. Wu, Y.; Yang, H.; Lin, Y.; Liu, H. Spatiotemporal Propagation Learning for Network-Wide Flight Delay Prediction. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 386–400. [\[CrossRef\]](#)
33. Zheng, H.; Wang, Z.; Zheng, C.; Wang, Y.; Fan, X.; Cong, W.; Hu, M. A graph multi-attention network for predicting airport delays. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *181*, 103375. [\[CrossRef\]](#)

34. Zhu, Q.; Chen, S.; Guo, T.; Lv, Y.; Du, W. A Spatio-Temporal Approach With Self-Corrective Causal Inference for Flight Delay Prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 20820–20831. [[CrossRef](#)]
35. Anuradha, G.; Vasana, M.B.; Yendluri, P. A Graph Neural Network Approach for Predicting Flight Delays and Inferring Delay Causes. In *Proceedings of the 2025 5th International Conference on Intelligent Technologies (CONIT)*; IEEE: New York, NY, USA, 2025; Volume 6, pp. 1–6. [[CrossRef](#)]
36. Franco, J.L.; Neto, M.V.M.; Verri, F.A.N.; Amancio, D.R. Graph machine learning for flight delay prediction due to holding manouver. *arXiv* **2025**, arXiv:2502.04233.
37. Ozcelik, E.; Germen, E.; Cetek, F.A. Prediction of Arrival Aircraft Time Spent in TMA with Graph Neural Network. In *Proceedings of the Integrated Communications, Navigation and Surveillance Conference, ICNS*; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2025. [[CrossRef](#)]
38. Liu, K.; Ding, K.; Cheng, X.; Chen, J.; Feng, S.; Lin, H.; Song, J.; Zhu, C. Airport Delay Prediction with Temporal Fusion Transformers. *arXiv* **2024**, arXiv:2405.08293.
39. Falque, T.; Mazure, B.; Tabia, K. Machine learning for predicting off-block delays: A case study at Paris—Charles de Gaulle International Airport. *Data Knowl. Eng.* **2024**, *152*, 102303. [[CrossRef](#)]
40. Patrickzel. Flight Delay and Cancellation Dataset (2019–2023). Kaggle Dataset. 2023. Available online: <https://www.kaggle.com/datasets/patrickzel/flight-delay-and-cancellation-dataset-2019-2023> (accessed on 2 February 2026).
41. Wandelt, S.; Chen, X.; Sun, X. Flight Delay Prediction: A Dissecting Review of Recent Studies Using Machine Learning. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 4283–4297. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.