Developing Novel Features Employed by Regression Methods to Solve Air Traffic Delay Prediction Problems

Loc Tran
Vietnam Aviation Academy, Vietnam
locth@vaa.edu.vn

Luong Minh Tri Nguyen
The University of Queensland, Australia
luongminhtri.nguyen@student.uq.edu.au

Huu Hiep Nguyen
Vietnam Aviation Academy, Vietnam
hiep_nh@vaa.edu.vn

Tan Phuc Tran
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
ttphuc@hcmut.edu.vn

Kim Anh Phan
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
pvkanh@hcmut.edu.vn

Linh Tran
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
linhtran@hcmut.edu.vn (corresponding author)

Received: 21 January 2026 | Revised: 21 February 2026 | Accepted: 28 February 2026

 DOI: https://doi.org/10.48084/etasr.17672

ABSTRACT

Flight delays are influenced not only by flight-specific factors but also by network effects, where disruptions at central airports propagate through downstream connections. This study proposes a lightweight, model-agnostic framework that injects network structure into standard flight-level regression for arrival delay prediction. Using Bureau of Transportation Statistics (BTS) on-time performance data (523,435 completed flights from 2025-01-01 to 2025-01-31), the study builds (i) a directed airport-route graph weighted by observed origin–destination counts and (ii) a rotation hypergraph in which each (tail number, day) forms a hyperedge connecting all airports visited, weighted by the number of legs. From these structures, four airport-level ranking signals are derived: Graph PageRank (GPR), GCN-Style Fixed-Point Ranking (GGCN), Hypergraph PageRank (HPR), and Hypergraph Fixed-Point Ranking (HGCN). Origin and destination ranking features are attached to each flight, and Linear Regression (LR), K-Nearest Neighbors (KNN) regression, and Multi-Layer Perceptron (MLP) regression models are evaluated under a time-aware split. Network-enriched features yield modest but consistent gains, with the best performance achieved by MLP using all ranking features.

Keywords-feature engineer; graph theory; PageRank; neural network; regression

I. INTRODUCTION

Flight delays are an operational challenge for airlines and airports, degrading passenger experience and inducing cascading disruptions across the air transportation system. Beyond local flight-specific factors, such as scheduled departure time, departure delay, and route distance, delays are influenced by network effects: congestion and disruptions at a few critical airports can propagate through downstream connections, amplifying delays system-wide. Large-scale operational records, such as the U.S. Department of Transportation / BTS on-time performance data, enable studying and predicting these phenomena at scale using data-driven methods [1].

A. Flight Delay Prediction with Machine Learning

Flight delay prediction is formulated as supervised regression or classification using flight-level tabular features such as schedule, carrier, airports, historical delays, distance, and sometimes weather and traffic variables. While ensembles and neural models can achieve strong predictive performance, the results are often sensitive to data availability, feature engineering, and operational context, particularly when models must generalize across time periods, carriers, or network conditions [2]. Deep learning models that explicitly capture temporal patterns (e.g., RNN/GRU variants and attention mechanisms) have also been explored to model non-linear and time-varying delay dynamics [3].

B. Graph-Based and Network-Wide Delay Modeling

Since delays propagate through airport and airline networks, delay prediction has been treated as a network learning problem. Graph-based models represent airports (or flights) as nodes connected by routes, traffic flows, geographic proximity, or learned dependencies. Spatiotemporal Graph Neural Networks (GNNs) and attention-based graph models have reported strong results for network-wide delay forecasting, highlighting the importance of explicitly capturing inter-airport dependencies and propagation effects [3]. However, many of these approaches require carefully engineered spatiotemporal inputs, specialized training pipelines, and higher computational cost. In practice, many operational settings still rely on flight-level records and lightweight regression models, where end-to-end spatiotemporal GNN pipelines may be harder to deploy or reproduce consistently.

C. Ranking-Based Network Features as Lightweight, Reusable Signals

A complementary perspective is to use network structure to derive compact, interpretable, and model-agnostic features that can be appended to standard regression pipelines. In network science, node importance measures such as PageRank summarize how influence flows through a directed graph and have been broadly used as centrality indicators in complex networks [4-6]. These centrality measures are appealing in aviation because airports serve as hubs within a directed traffic network, and hub-level disruptions can disproportionately contribute to downstream delays. However, centrality is often treated descriptively or embedded implicitly in large end-to-end GNN architectures, rather than being operationalized as explicit, reusable features that enrich flight-level datasets for downstream regressors.

This motivates a practical question: can network modeling and flight-level regression be bridged by computing graph-theoretic ranking signals (centrality) [4-6] and learning-based ranking signals (GCN-style propagation) [7-10], and then using them as additional predictors for standard regression methods? The present study adopts this strategy by computing ranking scores over (i) an airport-route graph and (ii) a rotation hypergraph, then attaches the resulting origin/destination ranking features to each flight record before training regression models for arrival delay prediction.

D. Higher-Order Operational Interactions via Hypergraphs

Airline operations also exhibit higher-order dependencies that are not naturally pairwise. A significant example is aircraft rotation: a tail number flies multiple legs within a day, coupling multiple airports through shared aircraft resources. Such multi-airport interactions can be represented naturally by hypergraphs, where a hyperedge connects more than two vertices. Classical hypergraph learning introduced normalized hypergraph operators and demonstrated their utility in clustering, embedding, and transductive learning [11, 12]. More recent hypergraph neural learning frameworks, such as HGNN and HyperGCN, extend message passing and convolution to higher-order relations [13-17]. Despite this progress, hypergraph modeling remains uncommon in practical flight delay prediction workflows, especially in a form that integrates cleanly with flight-level regression. Rotation-aware hypergraphs provide a principled way to encode resource coupling and multi-leg operational structure, complementing route-level connectivity.

E. Research Gaps

Three research gaps have been identified: (i) a feature-level integration gap, where network structure is often ignored by flight-level regressors and rarely injected explicitly as reusable ranking features [4-10]; (ii) a higher-order operational structure gap, where multi-leg resource coupling (e.g., rotations) is central to delay propagation but hypergraph learning remains underused in this context [11-17]; and (iii) a deployability and interpretability gap, since many spatiotemporal GNN approaches can be accurate but complex, whereas ranking-based features are lightweight and easy to append to tabular ML workflows.

F. Proposed Approach and Contributions

To address these gaps, the study proposes a unified framework that computes multiple ranking signals over graph and hypergraph representations of the air traffic system and injects them into a flight-level regression pipeline for arrival delay prediction. Specifically, the proposed framework:

- Constructs a directed airport graph from origin–destination movements and computes column-stochastic PageRank scores (with dangling-mass correction) as a topology-driven importance measure [4-6].

- Learns a GCN-style ranking on the airport graph via fixed-point propagation inspired by graph convolutional learning [7-10].
- Builds a rotation hypergraph where each tail number and day defines a hyperedge connecting all airports visited in that rotation, then computes HPR using normalized hypergraph random-walk operators [18-21].
- Learns a hypergraph convolutional ranking score via fixed-point propagation on a symmetric hypergraph operator, conceptually aligned with hypergraph neural learning [13-17].
- Attaches these ranking scores as origin/destination features to each flight record to create a network-enriched dataset that can be used by standard regression models, including LR, KNN regression, and MLP regression, for comparison against baseline tabular prediction.

Overall, the study develops a model-agnostic, interpretable, and operationally meaningful mechanism for injecting network structure—both pairwise (graph) and higher-order (hypergraph)—into practical flight delay prediction pipelines.

II. GRAPH-BASED RANKING FEATURES FOR FLIGHT-LEVEL REGRESSION

An airport-route graph was constructed from flight records, and two graph-based ranking signals, PageRank and a GGCN, were derived and later attached to each flight as origin/destination features.

A. Airport-Route Graph Construction

Let $\mathcal{V}$ denote the set of airports appearing as either origin or destination in the dataset within a chosen time window. A directed weighted graph $G = (\mathcal{V}, \mathcal{E})$ was constructed, where an edge $i \rightarrow j$ exists if at least one flight departs from the airport i and arrives at the airport j. The edge weight reflects observed traffic volume:

$$A_{ij} = \#\{flight\ with\ ORIGIN = i, DEST = j\} \quad (1)$$

yielding a sparse adjacency matrix $A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. This construction captures the empirical route network and naturally supports downstream random-walk ranking and diffusion-style propagation.

To define a random walk, the row-stochastic transition matrix was computed:

$$P_{row} = D_{\text{out}}^{-1}A, (D_{\text{out}})_{ii} = \sum_j A_{ij} \quad (2)$$

with standard safeguards applied for zero out-degree nodes. In the implementation, a column-stochastic form was used for PageRank by transposing:

$$P_{col} = P_{row}^T \quad (3)$$

so that probability mass flows via matrix-vector multiplication $P_{\text{col}}\, r$, complying with the classical PageRank update.

B. Column-Stochastic Pagerank on the Airport Graph

PageRank assigns each node a score proportional to the stationary probability of a random surfer who mostly follows links but occasionally teleports. Formally, for the damping factor $\alpha \in (0,1)$ and teleport distribution v (uniform in the baseline), PageRank solves the fixed-point equation $r = \alpha P_{\text{col}} r + (1-\alpha)v$, with practical handling of dangling nodes (columns of zeros) by redistributing their mass through v. This is the standard PageRank idea introduced in [4].

Let $\mathcal{D}$ be the set of dangling indices and define the dangling mass at iteration k as $m_k = \sum_{j\in\mathcal{D}} r_k[j]$. The column-stochastic iteration is then given by:

$$r_{k+1} = \alpha(P_{col} r_k + m_k\, v) + (1-\alpha)\, v, \quad \text{Followed by:}$$

$$r_{k+1} \leftarrow \frac{r_{k+1}}{1^\top r_{k+1}} \quad (4)$$

This update was run until convergence in ℓ_1 norm. Intuitively, airports that receive traffic from other important airports (directly or through multi-hop routes) obtain higher PageRank scores, providing an interpretable proxy for network prominence and potential disruption influence.

C. GCN-Style Fixed-Point Ranking (Learned Diffusion Score)

While PageRank provides a principled, parameter-free centrality score, it imposes a specific random-walk model. To obtain a complementary learned ranking signal, a lightweight GCN-inspired fixed-point ranker based on normalized graph diffusion, as popularized by GCNs [7-10], is introduced.

1) Symmetric Propagation Operator

First, a symmetric diffusion matrix suitable for stable feature propagation is built. Let $A_{\text{sym}} = A + A^\top$ and add self-loops I, then define:

$$\tilde{A} = A_{\text{sym}} + I, S = \tilde{D}^{-\frac{1}{2}}\, \tilde{A}\, \tilde{D}^{-\frac{1}{2}}, (\tilde{D})_{ii} = \sum_j \tilde{A}_{ij} \quad (5)$$

Operator S mixes local neighborhoods while preserving a normalized scale, analogous to the propagation core of GCN-style models [7-10].

2) Fixed-Point Ranker Architecture

A nonnegative ranking vector $r \in \mathbb{R}^{|\mathcal{V}|}$ with $\sum_i r_i = 1$ is maintained. The model computes a new ranking $r_{\text{new}} = f_\theta(r; S)$ by two rounds of propagation with a small MLP:

$$h_1 = \sigma(Sr),\ z_1 = \sigma(W_1 h_1 + b_1),\ h_2 = \sigma(S z_1),$$

$$u = W_2 h_2 + b_2, r_{new} = \frac{softplus(u)}{1^T softplus(u)} \quad (6)$$

where $\sigma(\cdot)$ is the ReLU and softplus$(\cdot)$ enforces positivity. The normalization ensures that r_{new} is a probability vector, comparable across datasets and time windows.

3) Self-Supervised Objective

Because airport "importance" labels are not directly available, θ is trained using a stability-seeking fixed-point criterion:

$$\mathcal{L}_{\text{fix}} = \| r_{\text{new}} - r \|_2^2 \quad (7)$$

encouraging the mapping to reach a consistent equilibrium. To avoid degenerate uniform solutions, a variance-promoting regularizer was added by maximizing $Var(r_{new})$:

$$\mathcal{L} = \mathcal{L}_{fix} - \lambda\, Var(r_{new}) \tag{8}$$

where $\lambda > 0$. The final learned score reflects a diffusion-consistent ranking shaped by the graph structure but not constrained to the exact random-walk assumptions of PageRank.

D. Feature Attachment to Flight Records

After computing airport-level rankings r^{GPR} (PageRank) and r^{GGCN} (GCN-style), these values are attached to each flight record as two origin/destination feature pairs:

$$\text{ORIG\_GPR} = r^{\mathrm{GPR}}(\text{ORIGIN}),\ \text{DEST\_GPR} = r^{\mathrm{GPR}}(\text{DEST}),\ \text{ORIG\_GGCN} = r^{\mathrm{GGCN}}(\text{ORIGIN}),\ \text{DEST\_GGCN} = r^{\mathrm{GGCN}}(\text{DEST}) \tag{9}$$

These features are then combined with baseline flight-level predictors such as scheduled departure time in min, departure delay, and distance to form a network-enriched tabular dataset for downstream regression.

III. HYPERGRAPH-BASED RANKING FEATURES FROM AIRCRAFT ROTATIONS

The air traffic system was modeled using a pairwise (route) graph. However, airline operations also exhibit higher-order coupling that is not naturally represented by edges. A significant example is aircraft rotation: the same tail number flies multiple legs in a day, linking several airports through shared aircraft resources. Delays can therefore propagate not only along routes but also along rotation-induced dependencies (e.g., late inbound aircraft affecting subsequent departures). To capture this structure, a rotation hypergraph was constructed, and two hypergraph-based ranking signals, HPR and a hypergraph convolution (fixed-point) ranking, were derived and then attached to each flight record as origin/destination features.

A. Rotation Hypergraph Construction

Let $\mathcal{V}$ denote the set of airports and $\mathcal{E}$ the set of hyperedges, where each hyperedge corresponds to a unique daily rotation identifier:

$$e \equiv (\text{TAIL\_NUM}, \text{DAY}) \tag{10}$$

For each flight in the dataset, its origin and destination airports were added to the corresponding hyperedge. Thus, a hyperedge $e \in \mathcal{E}$ connects all airports visited by that aircraft on that day:

$$e = \left\{ a \in \mathcal{V} \;\middle|\; \begin{array}{c} a \textit{ appears as ORIGIN or} \\ \textit{DEST for rotation (tail,day)} \end{array} \right\} \tag{11}$$

The hypergraph was represented using an incidence matrix $H \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$, and $H(v, e)$ is given by:

$$H(v,e) = \begin{cases} 1, & if\ v \in e \\ 0, & otherwise \end{cases} \tag{12}$$

To reflect operational intensity, each hyperedge was assigned a weight $w_e > 0$. In the proposed baseline, w_e equals the number of legs flown in that rotation (i.e., the count of flights associated with (tail,day). This weighting emphasizes rotations that involve more flight legs, and thus potentially more opportunities for delay propagation.

In addition, the hyperedge degree (size) and the vertex degree were defined using:

$$\delta(e) = |\, e\, | \tag{13}$$

$$d(v) = \textstyle\sum_{e\in\mathcal{E}} w_e\ H(v,e) \tag{14}$$

where $d(v)$ grows when an airport participates in many (or heavily weighted) rotations, capturing aircraft resource coupling around that airport.

B. Zhou-Style Hypergraph Operators and Hypergraph Pagerank

The classical normalized hypergraph framework based on the incidence matrix and diagonal degree/weight matrices was adopted:

$$\begin{aligned} W &= \mathrm{diag}(w_e) \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}, \\ D_e &= \mathrm{diag}\big(\delta(e)\big) \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}, \\ D_v &= \mathrm{diag}\big(d(v)\big) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|} \end{aligned} \tag{15}$$

A key intermediate matrix is the induced vertex–vertex affinity (a clique-like projection that remains properly normalized):

$$A_h = H\, W\, D_e^{-1}\, H^\top \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|} \tag{16}$$

From this, a row-stochastic hypergraph random-walk matrix and a symmetric propagation operator were defined as:

$$P_{\mathrm{row}}^{(h)} = D_v^{-1} A_h = D_v^{-1} H W D_e^{-1} H^\top \tag{17}$$

$$S_h = D_v^{-\frac{1}{2}} A_h D_v^{-\frac{1}{2}} = D_v^{-\frac{1}{2}} H W D_e^{-1} H^\top D_v^{-\frac{1}{2}} \tag{18}$$

For PageRank, the column-stochastic form $P_{\mathrm{col}}^{(h)} = (P_{\mathrm{row}}^{(h)})^\top$ was used. HPR was then computed via the same teleporting fixed-point iteration as discussed above:

$$r_{k+1} = \alpha\big(P_{\mathrm{col}}^{(h)}\, r_k + m_k\, v\big) + (1-\alpha)v \tag{19}$$

where v is the teleport distribution (uniform by default) and m_k redistributes mass from dangling vertices. In the hypergraph setting, dangling vertices correspond to $d(v) = 0$ (no incident hyperedges), and m_k is defined as $m_k = \sum_{v:d(v)=0} r_k[v]$. The result r^{HPR} assigns higher scores to airports that are repeatedly co-involved with many other airports within heavily weighted rotations, reflecting higher-order operational coupling.

C. Hypergraph Convolution (Fixed-Point) Ranking

HPR provides an interpretable, random-walk-based ranking, but it is still constrained by that specific diffusion mechanism. To obtain a complementary learned signal, a hypergraph convolutional fixed-point ranker is defined by reusing the same architecture as the graph case, replacing the graph propagation matrix S with the hypergraph operator S_h.

A probability ranking vector $r \in \mathbb{R}^{|\mathcal{V}|}$ is maintained, and the following update is computed:

$$r_{new} = f_\theta(r; S_h) \tag{20}$$

using two rounds of propagation and a small MLP, with positivity enforced by $\mathrm{softplus}(\cdot)$ and normalization to $\sum_i r_i = 1$. Training is self-supervised via a stability (fixed-point) objective with a variance-promoting term:

$$\mathcal{L} = \| r_{new} - r \|_2^2 - \lambda Var(r_{new}), \lambda > 0 \quad (21)$$

The resulting score r^{HGCN} can be interpreted as a diffusion-consistent importance measure under higher-order (rotation-based) connectivity, providing a learned alternative to HPR.

D. Feature Attachment to Flight Records

After computing airport-level hypergraph rankings r^{HPR} and r^{HGCN} these scores are attached to each flight as origin/destination features:

$$\mathrm{ORIG\_HPR} = r^{\mathrm{HPR}}(\mathrm{ORIGIN}), \mathrm{DEST\_HPR} = r^{\mathrm{HPR}}(\mathrm{DEST}), \mathrm{ORIG\_HGCN} = r^{\mathrm{HGCN}}(\mathrm{ORIGIN}), \mathrm{DEST\_HGCN} = r^{\mathrm{HGCN}}(\mathrm{DEST}) \quad (22)$$

Together with the graph-based features, these hypergraph-derived features provide a compact representation of higher-order operational dependence that can be directly fed into standard regression models. Experiments evaluate whether these network-enriched features improve arrival delay prediction compared to baseline flight-level predictors alone.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

1) Dataset and Preprocessing

The study uses the BTS On-Time Performance dataset, accessed through the BTS TranStats portal [1]. The flight-level operational records were retrieved, and the selected fields were exported to a local CSV file (*T_ONTIME_REPORTING_loc.csv*) for the period 2025-01-01 to 2025-01-31. After removing cancelled flights and applying basic cleaning (valid tail number, valid origin/destination airport codes, and numeric arrival delay), the resulting sample contains 523,435 completed flights. The target variable is the arrival delay (ARR_DELAY), measured in s. ARR_DELAY is converted from min to s by multiplying by 60. Baseline predictors include CRS_DEP_MIN, DEP_DELAY, and DISTANCE, where CRS_DEP_MIN is computed from CRS_DEP_TIME (HHMM) as min since midnight, and DEP_DELAY is used in its original unit as loaded from the CSV.

BTS On-Time Performance dataset [1] was selected because it is a large-scale, public, and widely used operational dataset that supports reproducible evaluation. It contains the key identifiers required by the proposed framework: ORIGIN/DEST (to build the directed route graph), TAIL_NUM and DAY/DATE (to define rotation hyperedges), and standard delay/schedule fields (to form baseline tabular predictors). While other aviation datasets exist (e.g., proprietary airline operation logs or airport surface/ATC datasets), they are often not publicly accessible or do not consistently provide tail-number rotation information at scale. The objective is to demonstrate a lightweight network-feature injection pipeline on a credible, reusable benchmark, with the same methodology applicable to other datasets when comparable fields are available.

B. Graph and Hypergraph Construction

From the cleaned records, the following graphs were constructed:

- A directed airport-route graph with $|V| = 329$ airports and ~5,647 nonzero weighted edges (origin–destination counts).
- A rotation hypergraph with $|V| = 329$ and $|E| = 142{,}311$ hyperedges, where each hyperedge corresponds to a (TAIL_NUM, DAY) rotation weighted by the number of legs.

From these structures, four airport-level ranking vectors were computed:

- GPR: column-stochastic PageRank with $\alpha = 0.85$, and dangling-mass correction.
- GGCN: fixed-point ranking obtained via normalized graph diffusion.
- HPR: column-stochastic PageRank on the hypergraph with $\alpha = 0.85$, and dangling-mass correction.
- HGCN: ranking obtained using hypergraph convolution–based fixed-point.

Each ranking is attached to flight records as origin/destination features (e.g., ORIG_GPR, DEST_GPR), forming feature sets: BASE, BASE+GPR, BASE+GGCN, BASE+HPR, BASE+HGCN, and BASE+ALL (containing all eight ranking features).

1) Train/Test Split

The dataset was split by day, with the first 80% of days for training and the remaining 20% for testing, avoiding leakage across flights from the same day.

C. Regression Models

Three standard regressors were evaluated under a unified preprocessing pipeline (standardization applied to all input features): (i) LR, (ii) KNN regression with $k = 15$ and distance-weighted averaging, and (iii) MLP regression with hidden layer sizes (128, 64), ReLU activations, Adam optimization, and early stopping.

D. Evaluation Metrics

The model performance was evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) on the test split (lower is better). Since ARR_DELAY is measured in s, MAE and RMSE are also in s. Table I presents the performance comparison on the test set using baseline and network-enriched features.

E. Discussion

Table I indicates that the benefit of network ranking features is regressor-dependent. For MLP, combining all ranking features (BASE+ALL) yields the best overall result (MAE = 592.179 s, RMSE = 804.183 s), improving over BASE-MLP (MAE = 598.910 s, RMSE = 809.068 s) by 6.731 s in MAE and 4.885 s in RMSE (approximately 1.12% and

0.60% reductions, respectively). For KNN, the strongest configuration is BASE+GPR, reducing RMSE from 932.962 s to 924.107 s (a reduction of 8.855 s, approximately 0.95%), consistent with PageRank acting as a compact global similarity signal in the standardized feature space. For LR, gains are modest when using all features, with BASE+ALL-LR reducing RMSE from 809.390 s to 807.734 s (a reduction of 1.656 s). Overall, the results support the central claim of the study: graph/hypergraph ranking signals provide lightweight, reusable predictors that can complement tabular covariates, with the largest benefit appearing when the downstream model can exploit nonlinear interactions (MLP) and when multiple ranking families are combined.

TABLE I. PERFORMANCE COMPARISON ON THE TEST SET USING BASELINE AND NETWORK-ENRICHED FEATURES

Feature set	Model	MAE (s)	RMSE (s)	Note
BASE	LR	599.539	809.39	
BASE	MLP	598.91	809.068	
BASE	KNN	667.436	932.962	
BASE+GPR	LR	599.505	809.411	
BASE+GPR	MLP	604.62	811.555	
BASE+GPR	KNN	655.226	924.107	Best KNN
BASE+GGCN	LR	599.539	809.39	
BASE+GGCN	MLP	593.87	806.17	
BASE+GGCN	KNN	667.436	932.962	
BASE+HPR	LR	599.412	809.455	
BASE+HPR	MLP	603.369	811.356	
BASE+HPR	KNN	655.95	925.773	
BASE+HGCN	LR	599.539	809.39	
BASE+HGCN	MLP	593.87	806.17	
BASE+HGCN	KNN	667.436	932.962	
BASE+ALL	LR	597.813	807.734	
BASE+ALL	MLP	592.179	804.183	Best overall
BASE+ALL	KNN	657.415	928.576	

The strongest configuration is BASE+ALL-MLP (MAE = 592.179 s, RMSE = 804.183 s). Compared with BASE-MLP, this configuration yields consistent gains: MAE decreases from 598.910 to 592.179, and RMSE decreases from 809.068 to 804.183. This supports the main hypothesis that compact network-derived rankings can add predictive signal beyond standard covariates. For KNN, the best result comes from GPR (BASE+GPR-KNN), improving over BASE-KNN, with RMSE decreasing from 932.962 to 924.107. This is consistent with KNN benefiting from additional geometry-defining features: ORIG_GPR/DEST_GPR act as global centrality summaries that help identify similar flights. HPR (BASE+HPR-KNN) provides a smaller improvement (RMSE 925.773), suggesting that rotation-based rankings are more smoothed and slightly less discriminative for this distance-based learner.

Although training logs for the fixed-point rankers appear degenerate (printed losses near 0), the downstream behavior is more nuanced: GGCN/HGCN do not change LR/KNN results, indicating that these features are effectively weak or redundant for simpler models, yet they do improve MLP (e.g., BASE+GGCN-MLP MAE = 593.870 s compared to the 598.910 s baseline). Overall, the best performance emerges when all ranking families are combined, suggesting complementary information even if individual families are not uniformly beneficial.

The proposed pipeline is feasible at scale, as demonstrated on a dataset containing 523k flights, 329 airports, and 142k hyperedges. Network ranking features yield measurable but regressor-dependent improvements, with the best result achieved by BASE+ALL-MLP. The fixed-point rankers likely require refinement (objective/constraints/diagnostics) to more reliably avoid near-trivial equilibria and to produce features that consistently benefit simpler regressors.

Strong performance has been reported using spatiotemporal deep models and graph-based learning for network-wide delay forecasting, including recurrent/GNN hybrids and attention-based spatiotemporal pipelines [2, 3]. The contribution of the present study is complementary: rather than training an end-to-end spatiotemporal predictor (often requiring engineered temporal tensors, weather/traffic exogenous inputs, and specialized training pipelines), the study provides a lightweight, model-agnostic mechanism that converts a route-graph and rotation-hypergraph structure into compact ranking features that can be appended to standard regression workflows. This design targets deployability and interpretability, enabling practitioners to enrich existing tabular pipelines with network information at low engineering cost. Direct numerical comparison with end-to-end spatiotemporal models in the literature is not always one-to-one due to differences in time horizons, feature availability (especially weather/traffic), and prediction targets; however, the results demonstrate that even without such heavy pipelines, network-derived rankings provide measurable gains and a practical bridge between network science and flight-level regression.

V. CONCLUSION

This study introduced a practical, model-agnostic framework for incorporating network structure into flight-level arrival delay prediction. Rather than relying on fully end-to-end spatiotemporal Graph Neural Network (GNN) pipelines, compact ranking signals are computed from two complementary representations of the air transportation system: a directed airport-route graph capturing observed traffic flows and a rotation hypergraph capturing higher-order operational coupling induced by aircraft daily rotations. From these structures, four airport-level ranking vectors are derived—Graph PageRank (GPR), a GCN-Style Fixed-Point Ranking (GGCN), a Hypergraph PageRank (HPR), and a Hypergraph Fixed-Point Ranking (HGCN)—and injected into standard regression models by attaching origin/destination ranking features to each flight record. This design preserves interpretability and deployability, since the resulting features can be appended to existing tabular ML workflows with minimal engineering overhead.

Experiments on 523,435 completed BTS flights from 2025-01-01 to 2025-01-31, using a time-aware day-based split, confirmed that network-enriched features can yield consistent (though modest) gains, and that the effect depends on the downstream regressor. The best overall performance was achieved by a Multi-Layer Perceptron (MLP) with all eight ranking features (BASE+ALL), improving over the baseline

MLP in both Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results also revealed a complementary pattern across model families: PageRank-based signals were most beneficial for the similarity-based K-Nearest Neighbors (KNN) regressor, while the learned fixed-point ranking signals were primarily exploited by the nonlinear MLP, with the greatest improvement emerging when all ranking families were combined. These findings support the main hypothesis that compact network-derived rankings can provide useful additional predictive signals beyond standard flight-level covariates such as scheduled departure time, departure delay, and distance.

Finally, the experiments reveal several directions for refinement. In particular, the fixed-point rankers showed near-degenerate training logs and limited impact on simpler regressors, suggesting that stronger constraints, improved objectives, or better diagnostics may be required to consistently avoid trivial equilibria and to produce rankings with more discriminative variation across airports. Future work can extend this framework by incorporating exogenous operational drivers (e.g., weather, capacity, traffic management constraints), exploring richer hyperedge definitions (multi-day rotations, fleet/airline-specific coupling), and evaluating robustness across longer time horizons and distribution shifts. Overall, the proposed ranking-based feature injection provides a lightweight bridge between network science (graph/hypergraph modeling) and practical flight-level delay prediction.

ACKNOWLEDGMENT

The authors would like to thank the Vietnam Aviation Academy, Ho Chi Minh City, Vietnam; HS Worms, Germany; Moscow Aviation Institute, Russia; and Vietnam Traffic Management Corporation, Vietnam, for their support.

The authors also acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

REFERENCES

[1] "Airline On-Time Performance Data (TranStats Database)." U.S. Department of Transportation, Bureau of Transportation Statistics, 2025, [Online]. Available: https://www.transtats.bts.gov/ontime/.

[2] J. Sun. et al, "Designing Recurrent and Graph Neural Networks to Predict Airport and Air Traffic Network Delays," in *10th International Conference for Research in Air Transportation FAA & Eurocontrol*, Tampa, FL, USA, Jun. 2020, pp. 1–8.

[3] X. Shen, J. Chen, and R. Yan, "A Spatial–Temporal Model for Network-wide Flight Delay Prediction Based on Federated Learning," *Applied Soft Computing*, vol. 154, Mar. 2024, Art. no. 111380, https://doi.org/10.1016/j.asoc.2024.111380.

[4] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, Apr. 1998, https://doi.org/10.1016/S0169-7552(98)00110-X.

[5] J. L. Franco, M. V. M. Neto, F. A. N. Verri, and D. R. Amancio, "Graph Machine Learning for Flight Delay Prediction Due to Holding Maneuver," *Physica A: Statistical Mechanics and its Applications*, vol. 685, Mar. 2026, Art. no. 131318, https://doi.org/10.1016/j.physa.2026.131318.

[6] A. N. Langville and C. D. Meyer, Eds., *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton, NJ, USA: Princeton University Press, 2012.

[7] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *5th International Conference on Learning Representations*, Toulon, France, Apr. 2017.

[8] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph Convolutional Networks: A Comprehensive Review," *Computational Social Networks*, vol. 6, no. 1, Dec. 2019, Art. no. 11, https://doi.org/10.1186/s40649-019-0069-y.

[9] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive Graph Convolutional Neural Networks," in Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, LA, USA, Feb. 2018, pp. 3546–3553.

[10] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability Methods for Graph Convolutional Neural Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Jun. 2019, pp. 10764–10773, https://doi.org/10.1109/CVPR.2019.01103.

[11] D. Zhou, J. Huang, and B. Schölkopf, "Learning with Hypergraphs: Clustering, Classification, and Embedding," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hofmann, Eds. Cambridge, MA, USA: The MIT Press, 2007, pp. 1601–1608.

[12] D. Zhou, J. Huang, and B. Schölkopf, "Beyond Pairwise Classification and Clustering Using Hypergraphs," Max Planck Institute for Biological Cybernetics, Tübingen, Germany, Technical Report TR-143, Aug. 2005.

[13] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3558–3565, Jul. 2019, https://doi.org/10.1609/aaai.v33i01.33013558.

[14] Y. Gao, Y. Feng, S. Ji, and R. Ji, "HGNN$^+$: General Hypergraph Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3181–3199, Mar. 2023, https://doi.org/10.1109/TPAMI.2022.3182052.

[15] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao, "Dynamic Hypergraph Neural Networks," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China, Aug. 2019, pp. 2635–2641, https://doi.org/10.24963/ijcai.2019/366.

[16] S. Kim, S. Y. Lee, Y. Gao, A. Antelmi, M. Polato, and K. Shin, "A Survey on Hypergraph Neural Networks: An In-Depth and Step-By-Step Guide," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Barcelona, Spain, Aug. 2024, pp. 6534–6544, https://doi.org/10.1145/3637528.3671457.

[17] D. Cai, M. Song, C. Sun, B. Zhang, S. Hong, and H. Li, "Hypergraph Structure Learning for Hypergraph Neural Networks," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, Vienna, Austria, Jul. 2022, pp. 1923–1929, https://doi.org/10.24963/ijcai.2022/267.

[18] Y. Takai, A. Miyauchi, M. Ikeda, and Y. Yoshida, "Hypergraph Clustering Based on PageRank," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug. 2020, vol. Virtual Event, pp. 1970–1978, https://doi.org/10.1145/3394486.3403248.

[19] K. Ameranis, A. DePavia, L. Orecchia, and E. Tani, "Fast Algorithms for Hypergraph PageRank with Applications to Semi-Supervised Learning," in *Proceedings of Machine Learning Research*, Vienna, Austria, Jul. 2024, vol. 235, pp. 1306–1330.

[20] L. Tran, T. Quan, and A. Mai, "PageRank algorithm for Directed Hypergraph." arXiv, 2019, https://doi.org/10.48550/ARXIV.1909.01132.

[21] Y. H. Piao, Y. J. Wang, and K. J. Li, "Identifying important nodes of hypergraph: An improved PageRank algorithm," *Chinese Physics B*, vol. 34, no. 4, Apr. 2025, Art. no. 048902, https://doi.org/10.1088/1674-1056/adb269.