## **Master Reference Document: A Hierarchical Framework for Causal Flight Delay Prediction**

### **1. Executive Summary**

This document outlines a comprehensive, multi-stage methodological framework for predicting flight arrival delays. Moving beyond the limitations of single-stage, correlational models, this approach aims to improve predictive accuracy and model explainability by explicitly modeling the intermediate causal mechanisms of delay formation.

The core of this framework is the development of "specialist models" that diagnose the probability of key intermediate problems—namely, **weather-induced National Aviation System (NAS) disruptions** and **cascading delays from late-arriving aircraft**. The outputs from these specialist models are then used as powerful, high-level features for a final integrated predictor. This hierarchical structure is designed to mirror the real-world chain of events, transforming the prediction task from a simple classification problem into a more robust, causally-aware diagnostic process.

### **2. Overall Methodological Framework**

Our approach is structured in three sequential phases:

1. **Data Foundation & Feature Engineering**: Transforming raw data into a rich, structured feature set.
2. **Causal Modeling of Intermediate Mechanisms**: Building specialist models to predict the probability of specific delay causes.
3. **Final Integrated Prediction**: Combining all information for a final, accurate, and explainable delay forecast.

### **Phase I: Data Foundation & Feature Engineering**

This foundational phase prepares the data and engineers a comprehensive set of features from multiple domains.

#### **A. Data Preparation and Sequencing**

* **Data Sources**: The project utilizes the **Airline On-Time Performance (AOTP)** dataset for flight records and the **Quality Controlled Local Climatalogical Data (QCLCD)** for hourly weather observations.
* **Preprocessing**: Raw data is cleaned by filtering out canceled and diverted flights1. A temporal-spatial join is performed to associate each flight with the 12-hour window of weather observations at its origin and destination airports2.
* **Aircraft Sequencing**: To model delay propagation, the dataset is processed to create event chains for each aircraft.
  + **How:** This is achieved by sorting the entire dataset first by Tail\_Number (the unique aircraft identifier) and then chronologically by scheduled\_departure\_time. Using a window function (shift()) within each Tail\_Number group, we link each flight to its predecessor's data, enabling the calculation of propagated risk features.

#### **B. Feature Engineering**

We will create three distinct categories of features for each flight.

**1. Flight & Temporal Features:**

* **How:** We utilize datetime accessors (e.g., .dt.hour, .dt.dayofweek) on the scheduled departure and arrival time columns to extract cyclical features. Spatial features like Origin\_Airport are transformed using a TargetEncoder to represent their historical delay propensity as a numerical value. TargetEncoder (1 for late, 0 for on-time) allows us to calculate how many flights departing from an airport have been late depending on a set threshold (e.g. 15 minutes): for instance if JFK had 10 flights and 3 were delayed then it will have a delay\_prop score of 0.3.

**2. Weather Aggregate Features:**

* **How:** After joining, the time-series weather data is aggregated using a groupby() operation on a unique flight identifier. Statistical functions (.agg(['mean', 'std', 'min', 'max'])) are applied to numerical variables. Trend features are calculated using a .diff() operation on time-sorted observations within each group. Wind vectors are computed using trigonometric functions (numpy.cos, numpy.sin) on the wind direction (converted to radians) and multiplied by wind speed.

**3. Expert-Driven Criteria (Domain-Specific Features):**

* **How:** These binary and ordinal features are created using conditional logic (e.g., numpy.where) and string matching operations.
  + The Flight\_Category is determined by applying a set of conditional thresholds for Visibility and cloud ceiling height (parsed from the SkyCondition string).
  + The Icing\_Risk\_Flag is created by combining a temperature range condition with string matching on the WeatherType descriptor for moisture-related codes (e.g., 'FG', 'RA', 'FZRA').
  + The Is\_Thunderstorm\_Activity flag is generated using a string containment check (e.g., .str.contains('TS')) on the WeatherType column.

### **Phase II: Causal Modeling of Intermediate Mechanisms**

This phase builds the core of our hierarchical system: two specialist models that diagnose the probability of specific delay causes.

#### **A. Specialist Model 1: Weather-Induced NAS Disruption**

* **Objective**: To predict the probability of a systemic air traffic disruption caused by weather.
* **Methodological Steps**:
  1. **Target Engineering**: A binary target variable, is\_significant\_nas\_delay, is created by applying a threshold of 15 minutes to the historical prorated NAS\_Delay\_Minutes column.
  2. **Training**: A Gradient Boosting Machine (GBM) classifier is trained on a balanced subset of the data. The training set is balanced using the **Random Under-Sampling** technique to ensure equal representation of both classes3.
  3. **Feature Generation**: The trained specialist model's .predict\_proba() method is used to predict the class probabilities for the *entire dataset* (both train and test sets). The probability of the positive class (1) is then stored as a new, high-value feature: Prob\_NAS\_Disruption.

#### **B. Specialist Model 2: Late-Arriving Aircraft (Domino Effect)**

* **Objective**: To predict the probability of a delay caused by the aircraft's late arrival from its previous flight.
* **Methodological Steps**:
  1. **Target Engineering**: A binary target is\_late\_arrival\_delay is created using a 15-minute threshold on the Late\_Aircraft\_Delay\_Minutes column.
  2. **Input Feature Creation**: The key input feature, Prob\_NAS\_Disruption(Previous\_Flight), is obtained by merging the Prob\_NAS\_Disruption feature from the previously sequenced flight.
  3. **Training & Feature Generation**: The process is identical to the NAS specialist model. A GBM is trained on a balanced dataset to learn the relationship between the predecessor's delay risk and the current flight's outcome. The output is a new feature for each flight: Prob\_Late\_Arrival.

### **Phase III: Final Integrated Prediction & Evaluation**

The final stage integrates all engineered information into a single predictor and evaluates its performance.

#### **A. Final Predictive Model**

* **Objective**: To classify a flight as "On-time" or "Delayed" based on a specified threshold (e.g., 30 minutes).
* **Final Feature Set**: The model is trained on a comprehensive set of inputs, including all Phase I features and the powerful probability scores (Prob\_NAS\_Disruption, Prob\_Late\_Arrival) generated in Phase II.
* **Model Algorithm**: We will use a final GBM classifier, selected for its high performance on heterogeneous, tabular data.

#### **B. Training and Evaluation Protocol**

* **How**:
  1. **Final Target & Balancing**: The final binary target (is\_Delayed) is created using the 30-minute arrival delay threshold. The training dataset for the final model is also balanced using Random Under-Sampling.
  2. **Model Fitting**: The final GBM classifier is trained on this balanced, comprehensive feature set.
  3. **Performance Evaluation**: The trained model's performance is assessed on a held-out, unseen test set. We will use sklearn.metrics functions (accuracy\_score, classification\_report, confusion\_matrix) to compute the final performance metrics. For direct and rigorous comparison with the source paper, these will be **Accuracy**, **On-time Recall ($Rec\_o$)**, and **Delayed Recall ($Rec\_d$)** 4.