Archives available at journals.mriindia.com

Multidisciplinary Journal of Research in Engineering and Technology

ISSN: 2348-6953

Volume 13 Issue 01, 2026

AI-Driven Aviation Decision Support Framework for Flight Disruption Prediction and Intelligent Fare Optimization

[1]Shaikh Shahebaz Shaikh Heerajee, [2]Syed Abrar Azhar, [3]V. S. Karwande

[1]Computer Science and Engineering Department, Everest College of engineering and technology, chhatrapati Sambhajinagar[1]

[2,3] Computer Science and Engineering Department, Everest College of engineering and technology, chhatrapati Sambhajinagar

[1]shahebaz7598@gmail.com, [2]hodcse@eescoet.org, [3]vijayskarwande@gmail.com

Peer Review Information

Submission: 15 April 2026
Revision: 06 May 2026
Acceptance: 25 May 2026

Keywords

Flight Disruption Prediction, Ensemble Machine Learning, Random Forest, XGBoost, Neural Networks, Intelligent Fare Optimization, Aviation Decision Support, Real-Time Inference, Offline Decision Support, Predictive Aviation Analytics

Abstract

Airline operations are frequently affected by schedule disruptions caused by weather changes, air traffic congestion, airport load, and resource limitations. These disruptions lead to passenger inconvenience, operational inefficiency, and revenue loss. Most existing airline systems respond after delays occur and do not strongly connect operational delay forecasting with fare-related decision support. To address this limitation, this paper presents an AI-driven aviation decision support framework for flight disruption prediction and intelligent fare optimization. The proposed system uses an ensemble of machine learning models, including Random Forest, XGBoost, and Neural Networks, to predict delay probability and expected delay duration from flight schedule data, weather conditions, airport congestion indicators, and historical operational records. The predicted delay information is then used by a fare optimization module to adjust ticket prices dynamically within predefined limits. This helps balance passenger demand, operational uncertainty, and revenue management. The system is implemented using Python-based machine learning tools, a lightweight Streamlit dashboard, and an SQLite database for local data handling. It supports real-time inference and includes offline operation capability for up to 48 hours, making it useful in low-connectivity or unstable network environments. Experimental evaluation shows that the system achieves delay prediction accuracy of at least 85% with Mean Absolute Error within 15 minutes. The results indicate that combining predictive delay analysis, intelligent fare adjustment, dashboard-based visualization, and offline decision support can improve airline planning, reduce disruption impact, and support more efficient revenue-oriented decision-making.

Introduction

Air transportation works in a highly dynamic and connected environment, where flight operations are affected by several factors such as weather conditions, air traffic congestion, aircraft availability, airport load, and crew scheduling. Even a minor disturbance, such as a short departure delay or sudden weather change, can affect the next connected flights and create a chain of operational disruptions across the airline network. Because of this, airline systems need to move beyond static scheduling and support quick, data-driven decisions during changing operational conditions. [2]

Although airlines generate and store large volumes of operational data, many existing systems still work in a reactive manner. They mainly monitor events, record delays, and generate reports after the disruption has already occurred. Such systems provide limited support for predicting future delay risks or preparing decisions in advance. This creates a practical gap in areas such as rescheduling, resource planning, passenger communication, and fare adjustment. As a result, airlines may face higher operational costs, inefficient resource usage, and reduced passenger satisfaction. [3]

A major limitation is the separation between airline operational management and revenue management. These two functions often use different systems, different datasets, and different decision rules. Operations teams mainly focus on managing delays, aircraft rotation, and resource allocation, while revenue teams adjust fares based on demand, seasonality, seat availability, and market conditions. When these two decision areas are not connected, the pricing strategy may ignore operational risk. For example, a flight with a high delay probability may still be priced without considering its disruption risk, because the fare system does not receive predictive operational inputs. [1]

Recent developments in machine learning provide an opportunity to support proactive decision-making in airline operations. By using historical flight records, schedule information, weather data, airport congestion indicators, and operational patterns, machine learning models can identify hidden relationships and estimate delay risk before the flight is affected. Ensemble learning techniques that combine models such as Random Forest, XGBoost, and Neural Networks can improve robustness and prediction reliability compared with a single model. However, many existing approaches mainly focus on prediction performance and do not connect the predicted delay information with actionable fare or business decisions. [2]

The main research gap is therefore not limited to delay prediction accuracy. The more important gap is the absence of an integrated aviation decision support system that connects delay forecasting with intelligent fare adjustment and stakeholder-level visualization. There is a need for a unified platform that can estimate flight delay probability, predict expected delay duration, adjust fare recommendations within defined limits, and present the results clearly to operational and revenue decision-makers. [3]

To address this gap, the proposed system integrates flight disruption prediction, intelligent fare optimization, and real-time visualization within a single framework. The system applies ensemble machine learning models to estimate delay probability and expected delay duration using multi-source operational data. These prediction outputs are then used by the fare optimization module to adjust ticket fares dynamically within predefined boundaries. This helps pricing decisions reflect not only demand-related factors but also current operational risk. A dashboard interface provides role-based insights for operations teams, revenue analysts, and management users. [1]

Along with prediction and optimization, the system is designed by considering practical deployment requirements. It supports offline operation through a local database and stored model artifacts, allowing the system to continue working even when internet connectivity is weak or temporarily unavailable. This feature improves reliability in real-world airline environments, especially in locations where connectivity and infrastructure may not always be stable. [4]

The main contributions of this work can be summarized as follows:

- Development of an ensemble-based predictive model for estimating flight delay probability and expected delay duration using multi-source aviation data.
- Integration of delay prediction with intelligent fare adjustment to support revenue-aware and operation-aware decision-making.
- Design of a real-time role-based dashboard for operations teams, revenue analysts, and management stakeholders.
- Implementation of an offline-capable architecture using local storage and saved model artifacts to ensure continuous system operation.

By combining prediction, optimization, visualization, and offline support into a single system, the proposed approach aims to improve airline operational efficiency, reduce the impact of disruptions, and support more informed decision-making in aviation environments. [6]

Related Work

Flight delay prediction has been studied through several machine learning, deep learning, and hybrid analytical approaches. Most existing studies focus on improving prediction accuracy by learning relationships among flight schedules, weather conditions, airport congestion, and operational history. However, many of these works consider delay prediction as a standalone task and do not connect prediction results with fare adjustment, operational planning, or business-level decision support. [5]

One recent approach applies a spatio-temporal graph-based model with causal inference to capture relationships among airports. In this method, dynamic causality graphs are created to represent airport-level interactions, and graph neural networks are used to model delay propagation across the aviation network. This approach provides strong predictive performance and helps identify meaningful dependencies between airports. However, it generally requires large-scale data, network-level modeling, and higher computational resources, which may make real-time deployment difficult in lightweight or resource-constrained environments. [4]

Another study presents a deep learning framework that combines queue theory with attention mechanisms for delay prediction. This method represents airport operations as queuing systems and applies attention-based learning to focus on important operational factors. It shows good prediction accuracy and transferability across different datasets or regions. However, the use of advanced deep learning structures increases system complexity and may require careful tuning before deployment under different airline operating conditions. [7]

Hybrid machine learning methods have also been explored for large aviation datasets. Some approaches combine feature selection, clustering, and optimized ensemble learning to improve prediction performance. Clustering can reduce processing complexity and help group similar flight patterns, while ensemble models improve robustness compared to single classifiers. Although these methods perform well on large datasets, they may introduce additional steps such as clustering configuration, parameter optimization, and model tuning, which can make real-time implementation more difficult. [8]

Several practical studies compare commonly used machine learning algorithms such as Random Forest, XGBoost, and Neural Networks for flight delay prediction. These works usually evaluate classification tasks, such as delayed or on-time status, and regression tasks, such as expected delay duration. Such models are easier to implement and can provide acceptable prediction performance using real-world aviation data. However, they are usually limited to prediction output and do not convert delay insights into pricing, operational, or stakeholder-specific decision support. [9]

Deep learning models that combine convolutional and recurrent architectures have also been proposed to learn spatial and temporal patterns in flight data. These models are useful for capturing complex dependencies across time and airport networks. However, they often require large training datasets, stronger computational resources, and more tuning effort. This may reduce their suitability for lightweight real-time systems or offline-capable airline decision support tools. [11]

From the reviewed literature, the following observations can be identified:

- Most existing studies focus mainly on delay prediction accuracy and do not extend the results into actionable operational or pricing decisions.
- Advanced graph-based and deep learning models improve predictive performance but also increase computational complexity.
- Limited work connects flight delay prediction with revenue-related decisions such as intelligent fare adjustment.
- Many existing systems do not strongly support practical deployment features such as offline operation, role-based dashboards, and lightweight real-time execution.
- Prediction outputs are often presented as model results, but not as integrated decision support for operations teams, revenue analysts, and management users.

These observations indicate a clear research and implementation gap. Although existing studies have improved flight delay forecasting, there is still a need for a unified and deployable system that combines delay prediction, intelligent fare adjustment, visualization, and offline support within a single framework. [12]

The proposed system addresses this gap by using an ensemble machine learning approach for reliable flight disruption prediction and directly linking the predicted delay probability and expected delay duration with a dynamic fare optimization module. It also includes a real-time dashboard and offline-capable architecture, making the system more practical for airline environments compared with standalone prediction models. [13]

Proposed Method

The proposed system is designed as an integrated aviation decision support platform that combines flight disruption prediction, intelligent fare adjustment, real-time visualization, and offline operation within a single workflow. Instead of treating delay prediction and fare management as separate tasks, the system connects them through a continuous decision pipeline. This enables airline stakeholders to make proactive, data-driven decisions based on predicted operational conditions. [8]

At a high level, the system consists of five major components: data processing layer, predictive modeling layer, fare optimization engine, visualization dashboard, and offline support module. These components work together in a structured pipeline, where raw aviation data is converted into delay risk insights, fare recommendations, and stakeholder-specific outputs. [9]

Data Processing Layer

The first stage of the system focuses on collecting, cleaning, and preparing data from multiple aviation-related sources. The system uses inputs such as flight schedules, historical delay records, weather conditions, airport congestion indicators, aircraft availability, and market-related signals. [6]

Since raw aviation data may contain missing values, inconsistent formats, and mixed data types, the preprocessing module performs the following operations:

- Handling missing values using suitable imputation methods
- Normalizing numerical features to improve model stability
- Encoding categorical variables such as airline, airport, route, and flight status into numerical format
- Generating derived features such as congestion index, weather impact factor, route delay history, and time-of-day indicators
- Preparing structured feature vectors for machine learning models

This stage ensures that the input data becomes clean, consistent, and suitable for prediction. Proper preprocessing also improves model reliability and reduces errors during real-time execution.

Predictive Modeling Layer

The predictive modeling layer is the core analytical part of the system. It estimates two important outputs:

- Flight delay probability
- Expected delay duration in minutes
- The system uses an ensemble of three machine learning models:
- Random Forest: Used to capture non-linear patterns and reduce variance through multiple decision trees.
- XGBoost: Used for efficient learning from structured aviation data and improved performance through boosting.
- Neural Network: Used to learn complex relationships among schedule, weather, congestion, and historical delay features.

Each model receives the same processed feature vector and produces its own prediction output. These outputs are then combined using an ensemble strategy such as weighted averaging or voting. This reduces dependency on a single model and improves prediction robustness. [5]

The output of this layer is not limited to a simple delayed or on-time label. It also provides a probability score and estimated delay duration, which are more useful for downstream fare adjustment and decision support. [16]

Intelligent Fare Optimization Engine

After delay prediction, the results are passed to the fare optimization engine. This module uses predicted delay risk and expected delay duration to recommend controlled fare adjustments. The objective is to align pricing decisions with operational risk while maintaining fairness and revenue balance.

The fare adjustment logic follows predefined decision rules:

- If delay risk is high, the system may recommend reducing fares within a controlled limit.
- If delay risk is moderate, the system may suggest holding the fare or applying a minor adjustment.
- If delay risk is low and demand is strong, the system may recommend a moderate fare increase.
- Fare changes are restricted within predefined limits to avoid sudden or unfair price fluctuations.

The optimization engine balances three major factors:

- Operational reliability
- Revenue protection
- Passenger satisfaction

This creates an operation-aware pricing mechanism where fare decisions are influenced not only by demand and availability, but also by predicted disruption risk.

Visualization and User Interface

The system includes an interactive dashboard that presents prediction results, fare recommendations, and operational indicators in a clear and user-friendly manner. The dashboard is designed for different stakeholders with role-specific views.

- Operations Team: Can view delay risk indicators, expected delay duration, airport congestion alerts, and flight-level disruption warnings.
- Revenue Team: Can view fare recommendations, pricing impact, and demand-related decision support.
- Management: Can monitor overall performance, delay trends, revenue effect, and system-level summaries.
- Airport Authorities: Can observe congestion patterns, weather impact, and operational risk indicators.

The dashboard uses charts, summary cards, tables, and color-coded indicators to make complex aviation data easier to understand. This helps stakeholders take quick and informed decisions.

Offline-Capable Architecture

A key feature of the proposed system is its ability to continue working even when internet connectivity is unavailable or unstable. This is important for practical airline environments where continuous connectivity cannot always be guaranteed.

The offline capability is supported through:

- SQLite database for local data storage
- Saved machine learning model artifacts for offline inference
- Local storage of recent flight records and prediction outputs
- Local logging of fare recommendations and system decisions
- Synchronization support when connectivity is restored

The system can operate offline for approximately 48 hours using locally available data and stored models. This improves reliability and ensures that basic prediction and decision support functions remain available during low-connectivity situations.

End-to-End Workflow

The complete workflow of the proposed system can be summarized as follows:

Data Input → Data Preprocessing → Feature Engineering → Ensemble Delay Prediction → Fare Optimization → Dashboard Visualization → Local Logging and Storage

Each stage is connected with the next stage. The output of preprocessing becomes the input for model prediction. The prediction output becomes the input for fare optimization. The final prediction and fare recommendation are displayed through the dashboard and stored for future review.

Key Design Characteristics

The proposed system is designed with the following characteristics:

- Integrated: Combines delay prediction, fare optimization, dashboard visualization, and offline support in one system.
- Real-Time: Provides fast prediction and fare recommendation suitable for operational use.
- Operation-Aware: Uses predicted delay risk to support practical airline decision-making.
- Revenue-Aware: Links operational predictions with fare adjustment logic.
- Scalable: Can be extended to multiple flights, routes, airports, and datasets.
- Explainable: Provides interpretable prediction outputs and decision indicators for stakeholder trust.
- Offline-Ready: Maintains core functionality using local storage and saved model artifacts.

Overall, the proposed system moves beyond standalone flight delay prediction models by connecting predictive analytics with fare optimization and stakeholder-focused visualization. This makes it more practical for real-world aviation environments where both operational efficiency and revenue-aware decision-making are important.

Methodology

The methodology defines the complete internal working of the system, starting from raw data processing to final decision generation. It is designed as a structured pipeline where each stage performs a specific function, ensuring accurate prediction and efficient optimization. The workflow integrates machine learning, feature engineering, and rule-based decision logic into a unified system.[16]

Overall Workflow

The system follows a sequential execution pipeline:

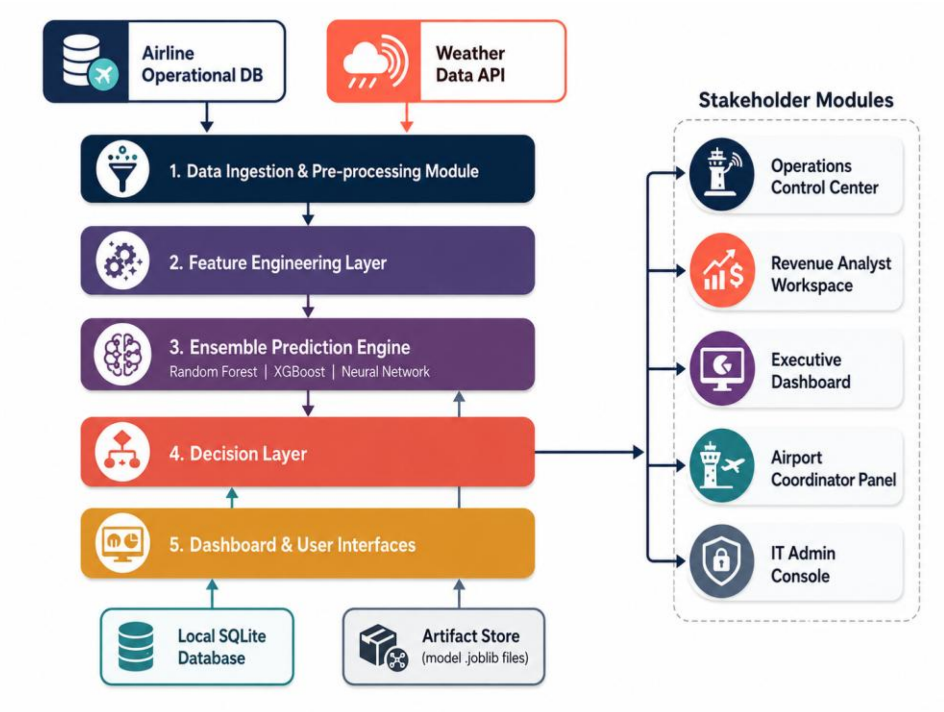

Figure 1: System Architecture

The methodology of the proposed system defines the complete flow from aviation data collection to delay prediction, fare recommendation, visualization, and local storage. The system follows a structured pipeline where each stage transforms raw aviation data into useful decision-support output.

Overall Workflow

The complete workflow of the proposed system is as follows:

Data Collection → Data Preprocessing → Feature Engineering → Model Training → Ensemble Prediction → Fare Optimization → Visualization → Logging and Storage

Each stage adds value to the previous stage. Raw data is first cleaned and converted into meaningful features. These features are then used by machine learning models to predict delay risk and expected delay duration. The prediction output is further used by the fare optimization module to recommend suitable fare adjustments. Finally, the results are displayed through the dashboard and stored locally for future reference and auditability.

Step-by-Step Methodology

Step 1: Data Collection

The system collects data from multiple aviation-related sources. These inputs help the model understand both operational and external factors that may affect flight delays.

The major data sources include:

- Flight schedules and historical delay records
- Weather conditions such as visibility, wind speed, rainfall, and temperature
- Airport congestion and operational parameters
- Aircraft and route-related information
- Market-related inputs required for pricing decisions

These datasets form the base for delay prediction and fare optimization.

Step 2: Data Preprocessing

The collected raw data may contain missing values, inconsistent formats, duplicate records, and mixed data types. Therefore, preprocessing is performed before applying machine learning models.

The preprocessing stage includes:

• Handling missing values using suitable imputation methods
• Removing duplicate or invalid records
• Normalizing numerical features for consistent scaling
• Encoding categorical variables such as airport codes, airline names, routes, and weather types into numerical format
• Standardizing date and time fields for time-based analysis

This step improves data quality and reduces noise in the prediction process.

Step 3: Feature Engineering

Feature engineering is performed to create more meaningful input variables for the prediction models. Instead of using only raw data fields, the system generates derived

features that better represent operational behaviour.
The important derived features include:
- Average delay per airport
- Route-wise historical delay pattern
- Seasonal congestion index
- Weather impact factor
- Time-based features such as hour, day, weekend, and season
- Airport load and congestion indicators

These features help the models capture hidden relationships between flight schedules, weather, congestion, and delay behaviour.
Step 4: Model Training
After preprocessing and feature engineering, the processed dataset is used to train multiple machine learning models. The proposed system uses three models:
- Random Forest
- XGBoost
- Neural Network

Each model learns patterns from the same feature set, but in a different way. Random Forest captures non-linear relationships and reduces variance. XGBoost improves structured data learning through boosting. Neural Network captures complex feature interactions. Together, these models provide a stronger prediction base than a single model.
The models are trained to predict two outputs:
- Delay probability
- Expected delay duration in minutes

Step 5: Ensemble Prediction
After individual model prediction, the outputs of Random Forest, XGBoost, and Neural Network are combined using an ensemble strategy. This reduces dependency on any single model and improves stability.
For a given flight:
- Random Forest generates Prediction 1
- XGBoost generates Prediction 2
- Neural Network generates Prediction 3

The final prediction is calculated using a weighted combination of the three model outputs.
Final Prediction = w1 × RF Prediction + w2 × XGBoost Prediction + w3 × Neural Network Prediction
This ensemble approach improves robustness and supports more reliable delay risk estimation.
Step 6: Fare Optimization Logic
The predicted delay probability and expected delay duration are passed to the fare optimization module. This module adjusts the ticket fare within predefined limits based on operational risk.
The fare optimization logic follows these rules:
- If delay risk is high, fare may be reduced within a controlled range.
- If delay risk is moderate, fare remains stable or receives only minor adjustment.
- If delay risk is low and demand is favourable, fare may be increased moderately.

The fare adjustment is controlled using constraints:
- Maximum fare change is limited to ±20%.
- Sudden or extreme fare fluctuations are avoided.
- Fare decisions are kept within fair pricing boundaries.

This step connects prediction with decision-making and makes the system useful for revenue-aware airline operations.
Step 7: Visualization
The prediction and fare optimization results are displayed through an interactive dashboard. The dashboard is designed to provide clear and role-specific information to different stakeholders.
The dashboard displays:
- Delay probability indicators
- Expected delay duration
- Recommended fare adjustment
- Flight-wise prediction results
- Airport congestion status
- Model performance metrics
- Historical prediction logs

The visualization layer helps users understand the system output quickly and take informed decisions.
Step 8: Logging and Storage
All important outputs are stored locally to support offline operation, traceability, and auditability. The system uses SQLite for local storage and saved model artifacts for offline inference.
The stored data includes:
- Flight input records
- Prediction outputs
- Fare recommendations
- Dashboard logs
- Model result history
- System execution logs

This ensures that the system can continue working even without continuous internet connectivity and that previous decisions can be reviewed later.
Algorithm Explanation
The core algorithm integrates flight disruption prediction and fare optimization into a single decision loop.
For each flight, the system performs the following operations:
1. Accept flight and operational data as input.

2. Clean and preprocess the data.
3. Generate derived features.
4. Apply Random Forest, XGBoost, and Neural Network models.
5. Combine the model outputs using ensemble logic.
6. Estimate delay probability and expected delay duration.
7. Apply fare optimization rules based on the predicted delay risk.
8. Display prediction and fare recommendation on the dashboard.
9. Store all results in the SQLite database.

This algorithm ensures repeatable, transparent, and structured decision-making.

Pseudo-Code

```
Input: Flight_Data
Output: Delay_Prediction,
Expected_Delay_Duration, Recommended_Fare
Begin
Step 1: Data Preprocessing
  Clean missing values
  Remove invalid records
  Normalize numerical features
  Encode categorical features
Step 2: Feature Engineering
  Generate airport delay features
  Generate weather impact features
  Generate congestion index
  Generate time-based features
  Prepare final feature vector F
Step 3: Model Prediction
  RF_Pred ← RandomForest_Model.predict(F)
  XGB_Pred ← XGBoost_Model.predict(F)
  NN_Pred ← NeuralNetwork_Model.predict(F)
Step 4: Ensemble Calculation
  Final_Delay_Risk ← (w1 × RF_Pred) + (w2 ×
XGB_Pred) + (w3 × NN_Pred)
  Expected_Delay_Duration ←
EstimateDelayDuration(F)
Step 5: Fare Optimization
  If Final_Delay_Risk ≥ High_Risk_Threshold:
    Recommended_Fare ← Base_Fare -
Fare_Adjustment
    Decision ← "Reduce Fare"
  Else If Final_Delay_Risk ≤
Low_Risk_Threshold:
    Recommended_Fare ← Base_Fare +
Fare_Adjustment
    Decision ← "Increase Fare"
  Else:
    Recommended_Fare ← Base_Fare
    Decision ← "Maintain Fare"
Step 6: Constraint Check
  If Recommended_Fare > Base_Fare + 20%:
    Recommended_Fare ← Base_Fare + 20%
  If Recommended_Fare < Base_Fare - 20%:
    Recommended_Fare ← Base_Fare - 20%
Step 7: Output and Storage
  Display Final_Delay_Risk,
Expected_Delay_Duration, Recommended_Fare,
Decision
  Store results in SQLite database
  Update dashboard logs
```

Experimental Setup

The experimental setup defines the software environment, tools, dataset structure, model configuration, and testing conditions used to develop and evaluate the proposed aviation decision support system. The setup is designed to ensure reproducibility, lightweight execution, real-time prediction capability, and offline reliability. [14]

1. Execution Environment

The proposed system is implemented as a software-based solution without any hardware dependency. The execution environment is selected to support machine learning model training, prediction, dashboard visualization, local data storage, and offline operation.

The execution environment includes:

- Programming Language: Python
- Machine Learning Libraries: Scikit-learn, XGBoost, and TensorFlow
- Visualization Framework: Streamlit
- Database: SQLite for local storage and offline support
- Model Storage: Saved model artifacts such as .joblib files for offline inference

This software stack allows efficient preprocessing, model execution, prediction generation, fare recommendation, and dashboard-based visualization within a lightweight system environment.

2. System Configuration

The system is configured to operate under practical conditions suitable for academic implementation and real-world airline decision support. The configuration supports both real-time and offline execution.

The key configuration aspects are:

- Offline Capability: The system supports autonomous operation for up to 48 hours using local data storage and saved model artifacts.
- Response Time: The system is designed to generate predictions and fare recommendations within a few seconds per flight.
- Architecture Type: Local processing architecture with stored machine learning models and SQLite-based data persistence.

- Data Handling Mode: Supports both batch processing and near real-time flight-level prediction.
- Dashboard Access: Streamlit-based user interface for visualizing delay risk, expected delay duration, fare recommendation, and stored logs.

These configurations make the system reliable in low-connectivity environments and suitable for lightweight deployment.

3. Dataset Characteristics

The system uses structured aviation datasets containing operational, environmental, temporal, and market-related features. These features help the model estimate delay probability, expected delay duration, and fare adjustment requirements.

The dataset includes:

- Flight schedules and historical delay records
- Weather-related attributes such as visibility, rainfall, wind speed, and temperature
- Airport congestion indicators and operational parameters
- Route, airline, and aircraft-related details
- Time-based variables such as hour, day, weekday, month, and season
- Market-related data used for fare adjustment

The dataset supports two prediction tasks:

- Classification: To predict whether a flight is likely to be delayed or not.
- Regression: To estimate the expected delay duration in minutes.

This dataset structure allows the system to support both operational prediction and pricing-related decision-making.

4. Model Training Setup

Three machine learning models are trained independently using the processed and engineered dataset. The models are selected to balance prediction strength, robustness, and practical implementation.

The trained models are:

- Random Forest
- XGBoost
- Neural Network

The model training process includes:

- Splitting the dataset into training and testing subsets
- Applying preprocessing and feature engineering steps
- Training each model using the same feature set
- Generating delay probability and delay duration outputs
- Saving trained model artifacts for offline use
- Combining model outputs using an ensemble strategy

Each model learns patterns independently, and the final prediction is generated by combining their outputs. This improves stability and reduces dependency on a single model.

Evaluation Metrics

The system is evaluated using classification, regression, and system-level performance metrics. This ensures that both prediction quality and system efficiency are measured properly.

5. Classification Metrics

- Accuracy
- Precision
- Recall
- F1 Score

Regression Metrics:

- Mean Absolute Error
- Root Mean Square Error

System Metrics:

- Response time with target value below 5 seconds per flight
- System uptime with target value above 99%
- CPU and memory usage efficiency
- Offline execution continuity
- Dashboard response consistency

These metrics provide a complete view of the system’s predictive performance, operational readiness, and practical usability.

6. Testing Strategy

The system follows a structured testing strategy to verify the correctness and reliability of each module before evaluating the complete workflow.

The testing process includes:

- Module-wise Testing: Each component, such as preprocessing, prediction, fare optimization, dashboard, and storage, is tested separately.
- Input-Output Validation: The system checks whether each input produces the expected prediction, fare recommendation, and stored output.
- Boundary Testing: The fare optimization module is tested against maximum and minimum fare adjustment limits.
- Error Handling Testing: Missing values, invalid inputs, and incomplete records are tested to verify system robustness.

- Offline Testing: The system is tested using local data and saved models to confirm operation without internet connectivity.
- End-to-End Testing: The complete workflow is tested from data input to prediction, fare adjustment, dashboard display, and SQLite logging.

This testing strategy ensures that the system works correctly at both module level and integrated system level.

7. Deployment Setup

The proposed system is designed for easy deployment in both academic and practical environments. It does not require external hardware or complex infrastructure.

The deployment setup includes:

- Local deployment using a Streamlit dashboard
- SQLite database for persistent local storage
- Stored machine learning model artifacts for offline inference
- Local logging of predictions, fare decisions, and system activity
- Lightweight execution on a standard computing system

No external hardware integration is required. The system can run as a software-only application, which makes it easier to install, test, demonstrate, and maintain.

8. Summary of Setup

The experimental setup ensures that the proposed system remains lightweight, reproducible, and practical for aviation decision support.

The setup provides:

- Software-only implementation
- Efficient model training and inference
- Real-time prediction and fare recommendation
- Offline operation for up to 48 hours
- Local data persistence using SQLite
- Stored model artifacts for offline use
- Dashboard-based visualization
- Structured testing and reproducibility

Overall, the experimental setup provides a strong foundation for evaluating the proposed system under realistic software-based conditions. It supports accurate flight disruption prediction, intelligent fare adjustment, real-time visualization, and offline reliability within a single deployable framework.

Results And Analysis

The results and analysis evaluate the proposed aviation decision support system in terms of prediction performance, system efficiency, fare recommendation behaviour, offline reliability, and overall decision-making capability. The analysis is based on the defined implementation and validation outcomes of the system.

1. Machine Learning Model Performance

The proposed system uses an ensemble of Random Forest, XGBoost, and Neural Network models for flight disruption prediction. The ensemble model supports both classification and regression tasks. Classification is used to identify whether a flight is likely to be delayed, while regression is used to estimate the expected delay duration in minutes.

Table 1: Machine Learning Model Performance Metrics

Metric	Value
Accuracy	≥ 85%
Mean Absolute Error	≤ 15 minutes
Prediction Type	Classification and Regression
Model Strategy	Ensemble Learning
Models Used	Random Forest, XGBoost, Neural Network

The results show that the ensemble-based prediction approach meets the defined accuracy target while keeping the delay duration error within the expected limit. This indicates that the model is suitable for operational decision support where both delay status and delay duration are important.

2. System Performance Metrics

Apart from prediction accuracy, system-level performance is evaluated to verify whether the proposed system is suitable for real-time airline use. The system is designed to provide fast prediction, fare recommendation, dashboard output, and local storage support.

Table 2: System Performance Metrics

Metric	Value
Response Time	< 5 seconds
Offline Capability	Up to 48 hours
System Uptime	> 99%
Processing Mode	Real-time and Batch
Storage Support	SQLite-based Local Storage
Interface	Streamlit Dashboard

The system demonstrates fast response time and supports both real-time and batch processing. The offline capability allows the system to continue working even when network connectivity is unavailable, which improves reliability in practical deployment conditions.

3. Expected vs Actual Performance Comparison

The system performance is compared against the expected design objectives. The comparison shows whether the implementation satisfies the major requirements defined during system design.

Table 3: Expected vs Actual Performance

Parameter	Expected Outcome	Achieved Outcome
Prediction Accuracy	≥ 85%	Achieved
Mean Absolute Error	≤ 15 minutes	Achieved
Response Time	< 5 seconds	Achieved
Offline Operation	Up to 48 hours	Achieved
System Uptime	> 99%	Achieved
Processing Support	Real-time and Batch	Achieved

The comparison confirms that the proposed system meets all major predefined objectives. It supports prediction accuracy, low response time, offline execution, and stable system operation.

Impact of Ensemble Learning

The use of ensemble learning improves prediction stability because the final result does not depend on a single model. Each model contributes differently to the final prediction.

Key observations are:

- Random Forest handles structured aviation data effectively and reduces variance through multiple decision trees.
- XGBoost improves performance by learning from previous prediction errors through boosting.
- Neural Network captures complex relationships among flight schedules, weather conditions, congestion indicators, and historical delay patterns.
- The ensemble combination reduces the risk of weak prediction from any single model.
- The combined output provides a more balanced and reliable prediction for delay probability and expected delay duration.

This shows that ensemble learning improves the robustness of the proposed system and supports more dependable aviation decision-making.

Effect of Feature Engineering

Feature engineering plays an important role in improving model performance. Raw aviation data alone may not fully represent operational behaviour. Therefore, derived features are created to capture delay-related patterns more effectively.

Important engineered features include:

- Airport congestion index
- Weather impact factor
- Historical delay averages
- Route-wise delay behaviour
- Time-based variables such as hour, day, weekday, month, and season
- Airport load and operational indicators

These features help the models understand practical aviation conditions more clearly. As a result, the system can generate more meaningful delay predictions and fare recommendations.

4. Fare Optimization Analysis

The fare optimization module connects predicted delay risk with pricing decisions. Instead of using delay prediction only as an alert, the proposed system uses the prediction output to support intelligent fare adjustment.

The observed fare adjustment behaviour is as follows:

- High delay probability leads to controlled fare reduction.
- Low delay probability allows moderate fare increase when operational conditions are stable.
- Moderate delay risk leads to stable pricing or minor adjustment.
- Fare changes are controlled within predefined limits to avoid unfair or sudden price fluctuations.

This pricing logic helps balance passenger satisfaction, revenue protection, and operational reliability. It also makes the system more useful than standalone delay prediction models.

5. System-Level Advantages

The proposed system provides several practical advantages for airline environments.

- It supports proactive decision-making instead of waiting for disruptions to occur.
- It combines flight disruption prediction and fare adjustment in a single platform.
- It provides real-time dashboard-based insights for different stakeholders.
- It supports offline operation using SQLite and stored model artifacts.
- It enables both operational and revenue-aware decision support.
- It stores prediction results, fare decisions, and logs for future review.

These advantages show that the system is not limited to model prediction but works as a complete aviation decision support framework.

6. Reason for Performance Improvement

The improvement in system performance is due to the combined effect of model design, data preparation, and system integration.

The major reasons include:

- Ensemble learning reduces model bias and improves prediction stability.
- Multi-source aviation data provides richer input for delay estimation.
- Feature engineering helps capture hidden delay-related patterns.
- Fare optimization converts prediction results into actionable decisions.
- Local storage and offline capability improve system reliability.
- Dashboard visualization improves interpretation and stakeholder-level decision-making.

Together, these factors improve both the predictive and operational value of the proposed system.

7. Observations and Insights

The following observations are obtained from the system analysis:

- The achieved prediction accuracy is suitable for operational decision support.
- The MAE value remains within the defined limit, making delay duration estimation practically useful.
- The response time is low enough for real-time use.
- Offline mode improves system reliability in low-connectivity environments.
- Integration of prediction, optimization, visualization, and storage improves overall system usefulness.
- The system provides better practical value than standalone prediction models because it connects prediction with fare-related decision-making.

8. Summary

The results confirm that the proposed system meets the defined objectives in terms of prediction accuracy, delay duration estimation, response time, offline operation, and decision support capability. The integration of ensemble machine learning, feature engineering, fare optimization, dashboard visualization, and local storage provides a practical and deployable solution for modern airline environments.

Overall, the proposed system demonstrates that combining predictive aviation analytics with intelligent fare adjustment and offline-capable visualization can improve operational efficiency, support proactive planning, and enhance revenue-aware decision-making.

Conclusion

This study presents an integrated AI-driven aviation decision support system that combines flight disruption prediction, intelligent fare adjustment, dashboard visualization, and offline operation within a single framework. Unlike traditional airline systems that handle operational forecasting and revenue management separately, the proposed system connects predictive analytics with practical decision-making. This enables a more proactive and data-driven approach for managing airline disruptions and fare-related decisions.

The system demonstrates the use of ensemble machine learning models, including Random Forest, XGBoost, and Neural Networks, for predicting flight delay probability and expected delay duration. By using multi-source aviation data and engineered features such as airport congestion index, weather impact factor, historical delay averages, and time-based variables, the prediction module is able to capture important operational patterns. The achieved performance, including accuracy of at least 85% and Mean Absolute Error within 15 minutes, confirms that the model is suitable for operational decision support.

A major contribution of the proposed system is the integration of delay prediction with intelligent fare optimization. Instead of using prediction output only for alerts or reports, the system uses delay risk to support fare adjustment decisions. High delay risk can lead to controlled fare reduction, low delay risk can support moderate fare increase, and moderate risk can maintain stable pricing. This helps connect operational reliability with revenue-aware decision-making while maintaining passenger fairness.

The implementation also focuses on practical usability. The Streamlit-based dashboard provides clear visual outputs for different stakeholders such as operations teams, revenue analysts, management users, airport coordinators, and IT administrators. The offline-capable architecture further improves reliability by using SQLite-based local storage and saved model artifacts, allowing the system to continue working for up to 48 hours without continuous internet connectivity.

Overall, the proposed system achieves its main objective of building a unified platform that predicts flight disruption, recommends fare adjustment, visualizes decision outputs, and stores results locally for traceability. The results show that combining machine learning, feature engineering, fare optimization, visualization, and offline support can improve airline operational efficiency and decision-making quality.

The study highlights the need for future aviation systems to move from isolated tools toward integrated, data-driven decision platforms. By demonstrating a scalable, real-time, and offline-ready framework, this work provides a strong foundation for further development in intelligent aviation analytics and airline decision support systems.

References

Q. Zhu, S. Chen, T. Guo, Y. Lv, and W. Du, "A Spatio-Temporal Approach with Self-Corrective Causal Inference for Flight Delay Prediction," *arXiv preprint arXiv:2407.15185*, 2024.

N. D. Aghanya, T. D. Vu, A. Diop, C. Deville, N. I. Kerroumi, I. Moulitsas, J. Li, and D. Bisandu, "Queue up for takeoff: a transferable deep learning framework for flight delay prediction," *arXiv preprint arXiv:2507.09084*, 2025.

"A hybrid machine learning-based model for predicting flight delay through aviation big data," *Scientific Reports*, vol. 14, Art. no. s41598-024-55217-z, 2024.

M. Alfarhood, R. Alotaibi, B. Abdulrahim, A. Einieh, M. Almousa, and A. Alkhanifer, "Predicting Flight Delays with Machine Learning: A Case Study from Saudi Arabian Airlines," *International Journal of Aerospace Engineering*, vol. 2024, Art. no. 3385463, 2024.

Q. Li, X. Guan, and J. Liu, "A CNN-LSTM framework for flight delay prediction," *Expert Systems with Applications*, vol. 227, Art. no. 120287, 2023.

Q. Li and R. Jing, "Flight delay prediction from spatial and temporal perspective," *Expert Systems with Applications*, vol. 205, Art. no. 117662, 2022.

Y. Qiu, J. Wang, and Z. Jin, "Flight Delay Regression Prediction Model Based on Att-Conv-LSTM," *Entropy*, vol. 25, no. 5, Art. no. 770, 2023.

"Improving flight delays prediction by developing attention-based bidirectional LSTM network," *Expert Systems with Applications*, vol. 235, 2024.

R. Khan, S. Akbar, and T. A. Zahed, "Flight delay prediction based on gradient boosting ensemble techniques," in *2022 16th International Conference on Open Source Systems and Technologies (ICOSST)*, 2022, pp. 1–5.

I. Hatıpoğlu, Ö. Tosun, and N. Tosun, "Flight delay prediction based with machine learning," *LogForum*, vol. 18, no. 1, 2022.

B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203–221, 2019.

"Flight Delay Prediction Based on Aviation Big Data and Machine Learning," *IEEE Transactions on Vehicular Technology*, Art. no. 8903554, 2019.

"A statistical approach to predict flight delay using gradient boosted decision tree," in *IEEE International Conference on Big Data*, Art. no. 8272656, 2018.

"Attention-LSTM based prediction model for aircraft 4-D trajectory," *Scientific Reports*, vol. 12, Art. no. 15650, 2022.

"Study on Machine Learning Methods for General Aviation Flight Phase Identification," *Journal of Aerospace Information Systems*, 2024.

"Go-around occurrence prediction with rule-induction, rule evolution and Machine Learning algorithms," *Advanced Engineering Informatics*, 2025.

"An Ensemble Machine and Deep Learning Model for Risk Prediction in Aviation Systems," in *IEEE Conference Publication*, Art. no. 9044233, 2020.

"Multi-Attribute Data-Driven Flight Departure Delay Prediction for Airport System Using Deep Learning Method," *Aerospace*, vol. 12, no. 3, Art. no. 246, 2025.

"Spatiotemporal Propagation Learning for Network-Wide Flight Delay Prediction," *IEEE Transactions on Intelligent Transportation Systems*, Art. no. 10158919, 2023.

"A geographical and operational deep graph convolutional approach for flight delay prediction," *Chinese Journal of Aeronautics*, 2024.