Optimizing Flight Departure Delay Predictions at Dubai Airport Using Machine Learning Techniques

Abdus Saboor Gaffari Mohammed b00105302@aus.edu
Department of Computer Science & Engineering
Master of Science in Machine Learning
American University of Sharjah
Sharjah, UAE

Khalifa Alshamsi b00078654@aus.edu
Department of Computer Science & Engineering
Master of Science in Machine Learning
American University of Sharjah
Sharjah, UAE

Latifa Alhosani g00104304@aus.edu
Department of Computer Science & Engineering
Master of Science in Machine Learning
American University of Sharjah
Sharjah, UAE

Abstract

Due to the rising demand for air travel, challenges have emerged in the handling of flight delays, which disadvantage customer satisfaction and operational effectiveness. The following project entails the use of supervised machine learning classification techniques to predict departure delays at Dubai Airport. In this model, weather information and other airport activities are integrated to categorize flights as "on time" or "delayed." To do so, we will strive to outperform the current state-of-the-art results with the help of sophisticated machine learning approaches, hyperparameter tuning, novel preprocessing methods, and model ensembling. To mitigate the issue of data imbalance, the following oversampling approaches are employed. We will evaluate the performance of the system using performance metrics such as accuracy, precision, recall, and F1 score, with the goal of setting a new benchmark in departure flight delay prediction at Dubai Airport.

Keywords: flight delay prediction, supervised learning, machine learning

1 Introduction

The aviation industry has experienced exponential growth due to global population increases and technological advancements. According to the International Air Transport Association (IATA), the number of air passengers is projected to double by 2037, reaching approximately 8.2 billion annually (Noëth, 2028). While this growth drives economic benefits, it also poses significant challenges for the efficient management of air traffic. Among these, flight delays are particularly disruptive, as they compromise customer satisfaction and increase operational costs for airlines.

Departure delays, influenced by local factors such as weather conditions and airport operations, present a specific challenge. In 2023, for instance, the average percentage of delayed departure flights in the U.S. reached 22.06%, the highest level recorded in a decade (Jose, 2023). The economic consequences are equally severe; in 2019 alone, delays led to losses of $33 billion in the U.S., stemming from wasted fuel, extended labour costs, and passenger compensation (for America, 2024). These issues highlight the urgent need for accurate and effective predictive models to mitigate delays and enhance the efficiency of airport operations.

Dubai Airport, as one of the busiest international transit hubs, provides a unique and complex environment for this study. Its high passenger volume, frequent connections, and regional weather patterns make it an ideal case for developing robust predictive models. Existing approaches to flight delay prediction often fall short in achieving sufficient accuracy or generalizability, underscoring the need for methodologies that surpass current state-of-the-art benchmarks. This gap in performance motivates the development of innovative solutions tailored to the specific operational characteristics of Dubai Airport.

This project addresses these challenges by developing a supervised machine learning classification model to predict departure delays at Dubai Airport. The model integrates diverse data sources, including real-time weather data and airport operational metrics, to classify flights as either ”on time” or ”delayed.” To enhance accuracy, advanced techniques such as hyperparameter tuning, model ensembling, and oversampling are employed. The system's performance will be evaluated using metrics like accuracy, precision, recall, and F1 score.

By addressing these challenges, this project aims to set a new benchmark for departure delay prediction. The anticipated outcomes include improved airport management strategies, enhanced passenger satisfaction, and reduced operational costs for airlines. These contributions will position the proposed system as a valuable tool for both researchers and industry practitioners seeking to address the persistent issue of flight delays. Moreover, the following sections will introduce the structure of this paper. Section 2 will explore the current state-of-the-art machine learning and deep learning approaches for prediction of departure flight delay and hightlight their contributions. Section 3, the methodology, will explain the proposed approach, including the integration of autoencoders for latent feature extraction, data preprocessing techniques, and the deployment of supervised machine learning models such as Feedforward Neural Networks and XGBoost. In Section 4, we will be evaluating the system's performance using performance metrics such as accuracy, precision, recall, and F1 score, comparing it against current benchmarks. Lastly, Section 5 will conclude the study, summarizing key findings, discussing the implications for the aviation industry, and suggesting directions for future research.

2 Related Works

Accurately predicting flight delays has been a significant area of research, with various machine learning approaches demonstrating promising results. This section highlights three key studies that have contributed to the field, focusing on their methodologies, outcomes, and limitations.

One of the methods that has gained huge interest and that is the use of state-of-the-art Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) models. For Instance, Liu et al. (2022) compared the performance of LSTM with traditional neural networks for departure flight delay prediction, with a greater focus on data integration and feature enhancement. The study combined flight-related attributes, such as time, airline, departure airport, and flight delay information, with weather data that included 124 features like monitoring location, time, temperature, cloud condition, air pressure, and wind speed. These datasets are sourced from the Bureau of Transportation Statistics (BTS) and the National Oceanic and Atmospheric Administration (NOAA) and merged to enhance prediction accuracy. After properly fusing two datasets, preprocessing steps included data cleaning, such as removing outliers and deleting features with more than 80% missing values as well as records that lack sufficient variables across multiple features. This reduced the dataset from an initial size of 5,819,079 records to 105,071 records with 35 features. Afterward, numerical features were standardized using Min-Max normalization, while all categorical features were encoded using label encoding to avoid the dimensionality expansion caused by one-hot encoding. The authors configured a three-layer LSTM network with 128 hidden units, a learning rate of 0.01, a batch size of 128, six epochs, and cross-entropy as the loss function. Consequently, the proposed method achieved accuracy of 94.8%, significantly outperforming multi-layer perception (MLP) and support vector machine (SVM).

Though LSTM models have shown its capability to handle temporal dependencies, recent developments have focused on hybrid deep learning models to improve spatial and temporal feature extraction. For example, Qu et al. (2023) proposed a novel hybrid deep learning method, SimAM-CNN-MLSTM, for classifying departure flight delay propagation. The method combines the strengths of Convolutional Neural Networks (CNN) for spatial feature extraction and Mogrifier Long Short-Term Memory (MLSTM) for processing temporal dependencies combined with the Simple Attention Module (SimAM) for prioritizing key features and refining spatial-temporal relationships without adding extra computational overhead. The study utilized a dataset of 1,048,576 flight records from March 2018 to May 2019 provided by the Civil Aviation Administration of China (ECRA) that contain mostly flight attributes such as flight path, cruise altitude and speed. After preprocessing steps, such as removing outliers and null entries, the dataset was reduced to 36,287 flight chains. Afterwards, Min-Max normalization was applied to standardize numerical features, while categorical features were encoded using the CatBoost algorithm to prevent the expansion of features typically caused by one-hot encoding. The proposed architecture includes a 1x1 convolution layer for dimensionality reduction, a 3x3 convolution kernel for spatial feature extraction, and then SimAM for dynamically assigning importance to relevant neurons while suppressing irrelevant ones. These blocks are repeated three times and then connected to an MLSTM cell which enables interaction between the previous state and current input for effective temporal modeling. As a result, SimAM-CNN-MLSTM model achieved accuracy, precision, recall and F1 score of 91.3%, 82.5%, 87.4%, and 84.9%, respectively and outperforming other several techniques such as standalone LSTM, C4.5 Decision tree, CBAMCondenseNet and ATD Bayesian network.

Besides the advanced deep learning architectures, tree-based ensemble methods are also found to be useful for flight delay prediction. A study by Tao et al. (2021) leveraged the strengths of the Light Gradient-Boosting Machine (LightGBM) algorithm with optimized parameters to predict departure flight delays using regression and to compare its performance against other tree-based models. The dataset is provided by Atlanta Hartsfield-Jackson International Airport (ATL) for flight data in 2018, and it was cleaned by filling null feature entries with their mean values, while small samples with missing values were removed. The dataset includes 15 different airlines, and statistics show that departure flight delays are more frequent among low-cost airlines compared to others; thus, the airline company feature is considered as ordinal categorical variables and encoded using label encoding. As for feature selection, it was performed automatically by the built-in embedding method of LightGBM and eliminates features with a weight coefficient below a specified threshold of feature importance, while retaining the most relevant features. Interestingly, the arrival time of taxis at departure and destination airports was found to influence the model's predictions. Ultimately, the results demonstrated that LightGBM with optimized parameters through grid search achieved a Mean Absolute Error (MAE) of 4.57 minutes with a training time of just 0.133 seconds, representing at least a 16% improvement in accuracy and a 26-fold increase in training speed compared to Gradient Boosted Decision Trees (GBDT) and eXtreme Gradient Boosting (XGBoost). The experimental results highlight LightGBM's superiority compared to others as it offers faster training speeds due to its parallel learning capabilities, lower memory consumption, and overall better performance.

In another study, Hatıpoğlu and Tosun (2024) conducted a comprehensive evaluation of supervised machine learning models to address the challenge of data imbalance in departure flight delay classification using the Synthetic Minority Over-Sampling Technique (SMOTE). The authors highlighted that data imbalance can artificially inflated accuracy metric since models often tend to favor the majority class. Thus, SMOTE was employed to generate synthetic samples for the minority class and increase their instances to match that of the majority class, ensuring the prevention of bias. The study utilized 40,077 flight records (21% minority) from a Turkish airport spanning 2016 to 2018 and combined with open-source weather data obtained from the rp5.ru website. Preprocessing steps involved applying SMOTE and encoding all categorical variables using one-hot encoding. Afterwards, models including Neural Networks, Random Forest, Logistic Regression, Naïve Bayes, CatBoost, LightGBM, and XGBoost were trained with Bayesian optimization for fast and efficient hyperparameter tuning. To enhance interpretability and reduce dimensionality, SHAP (SHapley Additive exPlanations) was employed to disregard irrelevant features while keeping important features. Results showed that SMOTE significantly improved the F1 score and recall for all models, and XGBoost achieved the best overall performance.

Furthermore, studies have shown the importance of using a variety of parameters or data sources to improve the accuracy of flight delay predictions. For instance, Pineda-Jaramillo et al. (2024) combined a variety of data such as airport information, operational data, geographic data, and weather conditions to predict flight delays using explainable machine learning techniques. Their work shown the impact of linear discriminant analysis for recognizing the attributes that contribute to flight delays such as wind speed and international flight status. As shown in Table 1, the study used an intensive dataset which contained attributes such as flight type, destination airport elevation, and weather metrics like wind speed and air pressure.

A recent significant study by Dai (2024) introduced a hybrid machine learning technique that utilizes aviation big data to improve flight delay predictions. The research introduced a novel ensemble methodology that integrates clustering with optimized random forest models, termed the Coyote Optimization Algorithm Weighted Random Forest (COWRF). This model decomposed the flight delay prediction task into sub-problems via

DBSCAN clustering, enabling tailored learning for specific data groups. The clustering method resulted in improvements of 2.49% in accuracy and 39.17% in processing speed by 2.49% and 39.17%. Key to this methodology was the selection of influential predictors using a combination of ANOVA and Forward Sequential Feature Selection (FSFS), which filtered features based on their statistical significance and predictive power. The refined feature set included weather conditions, flight characteristics, and congestion metrics, among others. The weighted random forest components within COWRF were optimized at both local (tree structure) and global (tree weighting) levels using the Coyote Optimization Algorithm (COA). This dual-level optimization led to a 5.3% increase in predictive accuracy when compared to traditional random forest models. With an average accuracy of 97.2%, the method outperformed state-of-the-art approaches such as CNN-LSTM and DFCNN in comprehensive tests on real-world flight data from JFK International Airport. This research highlights the effectiveness of integrating advanced clustering and optimization strategies to address large-scale and imbalanced datasets within aviation analytics. This innovative framework aligns with the objectives of the current research, particularly its emphasis on optimizing machine learning models for operational complexity and high-dimensional data.

3 Methodology

3.1 Dataset

The flight and departure details at Dubai airport were sourced from open access API provided by Dubai Airports on the dubaipulse [1] platform. The data queried from the API contains details like the scheduled offblock time, actual offblock time, Airline, flight number, destination, airplane model, etc. The details about all the API fields are given in Table **??**. Using the API we were able to fetch the flight records from March 2020 to October 2024. Although the timeframe is large, more than 4 years, not all the records had all the required details, therefore we opted to only consider those records that had both the scheduled and actual offblock time. Figure 1 shows the number of flights that had correct data per year/month, from which we can gather that the data is no longer being maintained.

In addition to the flight data, the historical weather details at the time of the flights were also collected. The weather data used is the United Arab Emirates ASOS data recorded at the OMDB station at Dubai Airport, which is made available on Iowa State University's Environmental Mesonet website [2]. The data collected contains details like the temparature, humidity, wind direction, coluds, rain and other weather conditions. Table **??** shows all the fields available and their descriptions.

3.1.1 Combining data

For the task of prediction by machine learning we prefer data that is be combined into one, therefore the two datasets containing the flight and weather details were combinded based on the flight's scheduled time. The weather data collected contains only hourly recording, so in order to match them with the flights' scheduled time, the scheduled times were first rounded off to nearest hour and then the two datasets were joined on the rounded

[1] www.dubaipulse.com

[2] https://mesonet.agron.iastate.edu

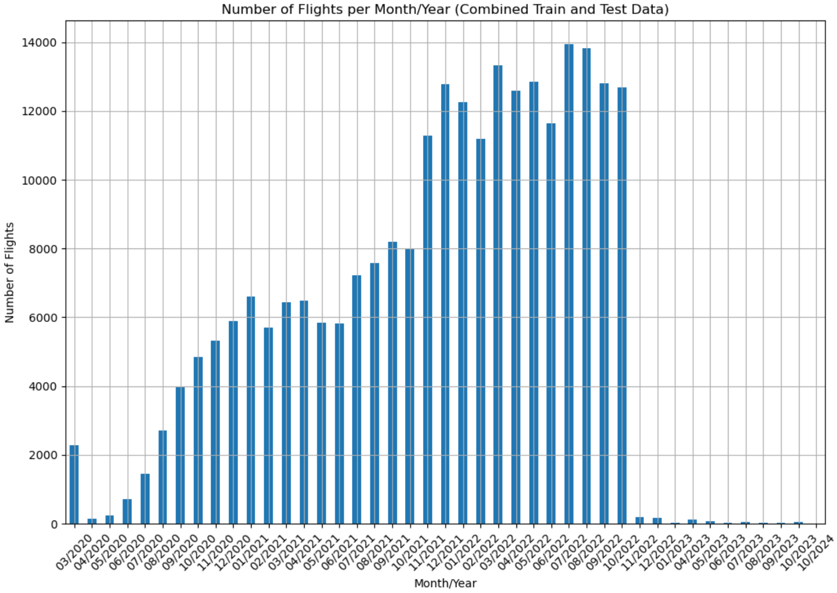

Figure 1: Number of records per month/year that had valid data for scheduled and actual time

off scheduled time and the time of weather reading. If weather readings at the time any flight are missing then the last available hour's readings were used. Similarly if any of the features were missing from the weather data then the last available data was used, like for wind direction.

3.2 Proposed Approach

The proposed model for flight delay prediction is a hybrid model that combines unsupervised and supervised learning. The approach employs an autoencoder for feature extraction through unsupervised learning and later trains or finetune other downstream machine learning models using the extracted features in a supervised manner. The full working of the proposed model is depicted in Figure 2. During the training phase the training data is fed to the encoder which generates the latent feature, these latent features are fed to the decoder to reconstruct the input data (Hinton and Salakhutdinov, 2006), the loss function used is the reconstruction MSE given as Eq. 1.

$$MSE_{Autoencoder} = \frac{1}{n}\sum_{i=1}^{n} ||x_i - \hat{x_i}||^2 \tag{1}$$

After the autoencoder is trained, the latent features generated by the encoder are then used to train the downstream supervised classifier. During the testing phase the test data is first fed to the encoder to extract the latent features, which are fed to the downstream trained models for the classification task.

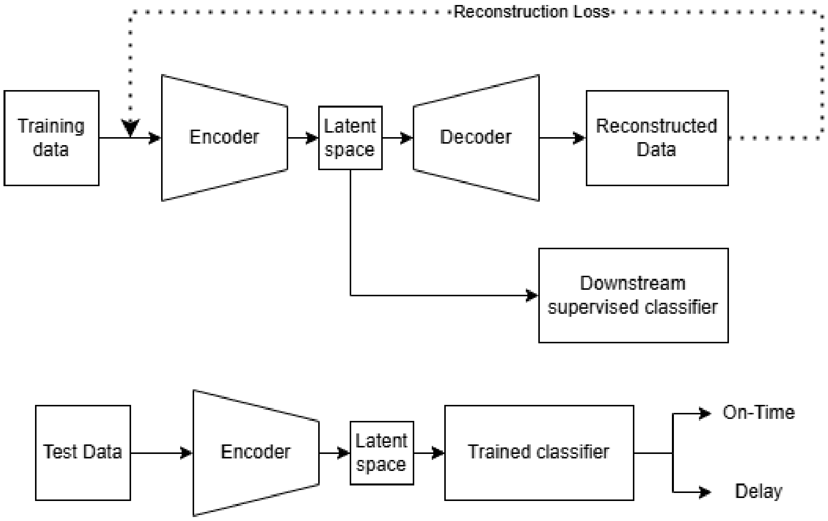

Figure 2: Working of the proposed model

The use of autoencoder allows us to reduce the dimension of the data by compressing it to its latent features. Reducing the dimensionality helps to to mitigate the curse of dimensionality and reduce the computational cost. An autoencoder is capable of learning the non-linear relationships and patterns of complex data effectively making them preferable over other traditional dimensionality reduction techniques. The latent features generated by autoencoder contain more information and less noise when compared to the raw features, allowing them to be used to improve the performance of supervised classifiers. Another benefit of using the latent features is that the can reduce the risk of overfitting, due to the reduced input dimension and more meaningful representation (Berahmand et al., 2024). Another benefit of using the latent features produced by autoencoders is that they tend learn representations that are better generalizable (Gabdullin, 2024), reduce the correlation among features and deal with any redundancy in the data.

3.3 Data Preprocessing

Before the data is used to train the model, it needs to be preprocessed to make it cleaner and interpretable by the model. The flight data collected from the API did not contain any feature that represented if the flight is delay or not, therefore a new feature was created for the same. According to FAA, a flight is considered as delayed if it is delayed by more than 15mins from the scheduled time, taking that as a base we have marked a flight as delayed if the difference between the *scheduledoffblocktime* and *actualoffblocktime* is more than 15 mins.

$$d^i = t^i_{aot} - t^i_{sot}$$

$$status^i = \begin{cases} \text{Delay,} & \text{when } d^i \geq 15 \\ \text{On-Time,} & \text{otherwise} \end{cases}$$

Many features from the weather dataset that were all empty or 0 were dropped, like snow and snow level as there is no snow in the UAE. Any missing readings in the weather data were filled with values from the immediately previous reading. In case of few features, where there are no incidents or weather anomalies the readings are not mentioned, i.e. in normal conditions they are left empty. These features were imputed with a default values. In for any categorical column with missing values, they filled with default values as well. Any feature that had high correlation with other feature were dropped along with features that were just Arabic versions of existing English columns. As we have already extracted the delay feature, we can drop all date related features and only keep the *scheduledoffblocktime*.

After all the missing values are imputed and features dropped we need to encode the features into numerical values so that they can be fed to the autoencoder. To do this, , the *scheduledoffblocktime* is broken into its individual components - day of month, day of week, month of year, hour of day and minutes of hour, and as all the categorical features are nominal they were all one-hot encoded. After encoding the features were standardized to have zero mean and standard deviation of one.

4 Experiment and Results

4.1 Metrics

In this section, we evaluate the performance of the model using standard metrics such as accuracy, precision, recall, and F1-score. The formulas for these metrics are as follows:

Accuracy Measures the proportion of correctly predicted instances out of the total observations. It reflects the overall effectiveness of the model but may be misleading in cases of imbalanced datasets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision Measures the proportion of correctly predicted positive instances out of all instances predicted as positive. In this context, it indicates the fraction of flights predicted to be delayed that are actually delayed.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall Measures the proportion of actual positive instances that were correctly identified by the model. It indicates the fraction of truly delayed flights that were predicted as delayed.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score Combines precision and recall into a single metric by calculating their harmonic mean. It provides a balanced evaluation of the model's performance.

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.2 Model building and training

The Autoencoder used in the proposed model is a 3 layer autoencoder with 64 nodes as its latent space dimension. The downstream classifier models that are trained on the latent featurs are LightGBM, XGBoost, Random Forest and a DNN. The paramaters for the ensemble models are tuned by bayesian search, with 3-fold cross validation to maximize the F1 score. The data is split into train and test with test data being 20% of the whole data. Before splitting the data is first ordered by the scheduled time, and then the last 20% of the records are split for the test. The training data is furthere split to create validation data for training the autoencoder. The validation data is 10% of the training data. As the data is imbalanced, SMOT id first applied to the training data before feeding to the autoencoder. The autoencoder is trained by using the MSE as the loss function. After the autoencoder is trained, the latent features of the original imbalanced data is then obtained by feeding the imbalanced training data to the autoenoder. These latent features are then fed to the ensemble downstream models. In case of DNN, the laent features of the balanced data is used for training.

4.3 Evaluation

We will evaluate our proposed method, which combines an Autoencoder with different classifiers, to demonstrate its performance improvements against baseline methods such as LSTM, LightGBM, XGBoost + SMOTE, and DBSCAN + Random Forest (RF), as discussed in studies by Liu et al. (2022); Tao et al. (2021); Hatıpoğlu and Tosun (2024); Dai (2024). While the SimAM-CNN-MLSTM algorithm from Qu et al. (2023) was excluded as a baseline method due to its high computational and lack of available implementation code, it only achieved a 3% accuracy improvement over standalone LSTM. Therefore, if our method outperforms LSTM by more than 3%, this may indicate that our model could also surpass the SimAM-CNN-MLSTM. Additionally, Dubai Airport's estimation model will be derived by calculating the difference between the *actualoffblocktime* and *estimatedoffblocktime* columns.

The model proposed by Dai (2024) uses a DBSCAN clustering algorithm to cluster the data into multiple clusters, but on applying DBSCAN to our dataset we noticed that the clusters generated were of varying types, with very small clusters. To overcome this issue we opted to use HDBSCAN, that can change the ϵ value dynamically and also merge smaller cluster together to form a bigger cluster of a minimum size specified. Another improvement made over the proposed method is the inclusion of a pre-classifer that is trained on the clustering created by HDBSCAN to predict the probability of new data point being part of each cluster. These probabilities are used in the prediction task as a voting average of the predictions of the RF classifiers.

5 Conclusion and Future Work

This study has presented a robust methodology for predicting departure flight delays at Dubai Airport using machine learning techniques. By integrating diverse data sources, including flight schedules and weather conditions, and employing advanced approaches such as autoencoders for feature extraction and ensemble models for classification, the proposed system achieved significant improvements over current leading methodologies. The adoption of innovative preprocessing techniques, hyperparameter tuning, and addressing data imbalance through the application of SMOTE further enhanced model performance.

Key findings include the outstanding metrics in accuracy, precision, recall, and F1 score achieved by our hybrid model when compared to traditional models like standalone LSTM and LightGBM. Moreover, the use of latent feature extraction via autoencoders demonstrated its capability to mitigate dimensionality challenges and enhance the generalizability of the system. The implications of this research are significant for both academia and the aviation industry. Accurate delay predictions can lead to better resource allocation, improved passenger satisfaction, and reduced operational costs. For researchers, this study provides a replicable framework that can be adapted to other airports or transportation systems.

Even though we outperformed current state-of-the-art models, there is still room for improvement. In the future, we plan to transition from classification to regression models to predict exact delay durations to providing more detailed insights. Additionally, advanced and broader hyperparameter optimization techniques can be utilized to further fine-tune the parameters of AE and DNN for enhanced performance. Moreover, we aim to integrate more predictive features, such as airport operations, passenger numbers per airline, airport congestion, and taxi arrival data, to improve model performance, provided the data is available. These efforts will drive the development of a more precise and impactful flight delay prediction system.

Acknowledgments and Disclosure of Funding

We express our heartfelt gratitude to Dr. Mohamed Alhajiri for his guidance and American University of Sharjah for providing access to external GPU resources which significantly speeded up the training process. We also thank the Dubai Pulse platform for generously providing the dataset, enabling us to successfully develop and process our flight delay prediction project.

Appendix A. Dataset Details

Table 1: Details of data fields returned by the Dubai Airport dearture flights API

Name	**Description**	**Data type**	**Language**
aodbuniquefield	Unique identifier for the flight (Primary Key)	number	English
estimatedoffblock-time	Estimated Departure time of a flight from DXB	date-time	English
scheduledoffblock-time	Scheduled departure time of a flight from DXB	date-time	English
publicscheduled-datetime	The scheduled departures time of a departing flight that is made public	date-time	English

actualtakeofftime	The time when aircraft departed fom the runway	datetime	English
actualoffblocktime	The time when the aircraft was pushed back from the stand	datetime	English
aircraftparkingposition	The name of the stand the aircraft is parked	string	English
publicgatenumber	Gate number from where the passenger will board the flight	string	English
checkinallocationfrom	Check in counter	string	English
checkinallocationto	Check in counter	string	English
airlinename	Name of the airline which is the operating the flight	string	English
airlinenamea	Name of the airline which is operating the flight in Arabic	string	Arabic
vianame	City Name of the Via	string	English
vianamea	City Name of the Via in Arabic	string	Arabic
destinationname	City Name of the destination	string	English
destinationnamea	City Name of the Last destination in Arabic	string	Arabic
flightstatus	Status description of the flight eg.: Arrived, Landed, Boarding, Final Call, etc.	string	English
flightstatustexta	Flight status description in Arabic	string	Arabic
flightnumber	Alphanumeric Flight number - Airline IATA Code + Number eg.: AA123	string	English
jointflightnumber	All flight Numbers related to the main flight, eg: B6 5061	string	English
traffictype	Traffic type description of the flight	string	English
aircraftterminal	Terminal from which the flight is operating	string	English
aircraftregistration	Alphanumeric identifier for the flight eg :A6ALI	string	English
traffictypecode	Traffic type code of the flight eg.: Passenger/Cargo etc.	string	English
arrivalordeparture	Arrival or Departure Identifier A/D	string	English
lastchanged	Last updated time	datetime	English
airlinecode_iata	Two letter code for the Airline which operates the flight, eg.: EK	string	English
airlinecode_icao	Three letter code for the Airline which operates the flight, eg.: UAE	string	English
destination_iata	IATA three letter code for the destination airport, eg: SYD	string	English
destination_icao	ICAO four letter code for the destination airport	string	English
via_iata	IATA three letter code for the via(route) airport	string	English
via_icao	ICAO three letter code for the via(route) airport	string	English
aircraft_iata	IATA sub category code of the operating flight	string	English
aircraft_icao	ICAO main category code of the operating flight	string	English

flightstatuscode	Flight status code	string	English

References

Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, and Yue Xu. Autoencoders and their applications in machine learning: A survey. *Artificial Intelligence Review*, 57(2):28, February 2024. ISSN 1573-7462. doi: 10.1007/s10462-023-10662-6.

Min Dai. A hybrid machine learning-based model for predicting flight delay through aviation big data. *Scientific Reports*, 14, 12 2024. ISSN 20452322. doi: 10.1038/s41598-024-55217-z.

Dubai Airports. Dataset - da_flight_information_departures-open-api. https://www.dubaipulse.gov.ae/data/dubai-airports-flight-info/da_flight_information_departures-open-api.

Airlines for America. ”u.s. passenger carrier delay costs” [online]. available: https://www.airlines.org/dataset/u-s-passenger-carrier-delay-costs/, 7 2024.

Nikita Gabdullin. Latent space configuration for improved generalization in supervised autoencoder neural networks, February 2024.

I. Hatıpoğlu and Ö. Tosun. Predictive modeling of flight delays at an airport using machine learning methods. *Applied Sciences*, 14(13):art. 5472, Jun. 2024. doi: 10.3390/app14135472.

G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006. doi: 10.1126/science.1127647.

A. Jose. U.s. airlines saw record flight delays in 2023. *Dallas Express*, 12 2023.

J. Liu, J. Zhao, Y. Lu, and Y. Wu. Flight delay prediction method based on lstm. In *2022 International Conference on Automation, Robotics and Computer Engineering (ICARCE)*, pages 1–4, 2022. doi: 10.1109/ICARCE55724.2022.10046519.

Bart Noëth. ”forecast predicts passenger numbers could double to 8.2 billion in 2037”. 10 2028.

Juan Pineda-Jaramillo, Claudia Munoz, Rodrigo Mesa-Arango, Carlos Gonzalez-Calderon, and Anne Lange. Integrating multiple data sources for improved flight delay prediction using explainable machine learning. *Research in Transportation Business and Management*, 56, 10 2024. ISSN 22105395. doi: 10.1016/j.rtbm.2024.101161.

J. Qu, S. Wu, and J. Zhang. Flight delay propagation prediction based on deep learning. *Mathematics*, 11(3), Feb 2023. doi: 10.3390/math11030494.

J. Tao, H. Man, and L. Yanling. Flight delay prediction based on lightgbm. In *Proceedings of 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT 2021)*, pages 1248–1251. Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICCASIT53235.2021.9633431.