Improving Flight Delay Prediction Using Integrated Departure Disruption, Airline-Route Stability, and Schedule Intensity Features

Huthaifa Aljawazneh
Department of Business Intelligence - Faculty of Business, Al-Zaytoonah University of Jordan, Amman, Jordan

***Abstract*—In the aviation industry, flight delays represent a major challenge due to their economic impact, operational disruptions, and adverse effects on passenger satisfaction and transportation efficiency. This study proposes an integrated feature engineering framework for flight delay prediction that combines three groups of engineered features, namely Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs), with the original flight delay dataset. These feature groups are designed to capture complementary operational, historical, and scheduling characteristics that are not directly represented in the original data. Five machine learning classifiers were evaluated across five experimental scenarios. These included the original feature set, the original feature set combined separately with each engineered feature group (DDFs, ARSFs, and SIFs), and the original feature set combined with all three engineered feature groups. This design enabled assessment of the individual and complementary contributions of the proposed features to flight delay prediction. The results indicated that the engineered features improved prediction performance across the evaluated configurations. In particular, combining all three engineered feature groups with the original feature set outperformed the baseline feature set and each individually combined feature-set configuration, supporting the complementary predictive value of the proposed engineered features. The complete feature set with LightGBM and SMOTE-Tomek achieved the best performance, with an accuracy of 0.9820 and an AUC of 0.9884, compared with 0.9487 and 0.9663, respectively, for the baseline feature set.**

Keywords*—*Flight delay prediction; feature engineering; aviation analytics; class imbalance; machine learning

I. Introduction

Flight delays are considered an important indicator of the performance of transportation systems [1], since any disruption at airports can spread through interconnected flights and affect the entire aviation network [2]. In the airline industry, a flight delay can be defined as the difference between the scheduled and actual departure/arrival time of an airplane. Moreover, such delays are often affected by preceding operational disruptions and congestion conditions. Since aircraft typically operate multiple flights during the day, delays in earlier segments may influence subsequent departures and arrivals [3].

According to the estimates made by the International Air Transport Association (IATA), global demand for air traffic is expected to increase by 3.7% annually to reach about 7.2 billion passengers by 2035, compared with 3.8 billion passengers in 2016. Additionally, data reported by the Civil Aviation Administration of China (CAAC) indicate that 47.46% of delays are caused by unfavorable weather conditions, 21.14% due to air routes, while 2.31% are attributed to airline operations and technical difficulties, and the remaining 29.09% are associated with air traffic control and other factors [3].

Furthermore, flight delays may also cause considerable economic consequences for passengers, airlines, and airport operations [1], with estimated direct and indirect costs totaling around 33 billion dollars within the National Aviation System (NAS) in 2007 [4]. In addition to financial losses, such delays may substantially increase passenger dissatisfaction, conflicts between passengers and airlines, and reduce the operational efficiency of the air transportation system [3]. Moreover, delays can impose additional operational costs on airlines and lead to higher fuel usage and environmental emissions [1].

Although many machine learning and deep learning approaches have been developed for flight delay prediction, most studies rely primarily on operational, weather, temporal, or network-related variables. In addition, while feature engineering has been shown to improve predictive performance, these feature categories are often examined separately. Few studies have investigated the combined effect of departure disruption indicators, airline-route stability characteristics, and schedule intensity measures within a single feature engineering framework. In addition, the relationship between these engineered features and class-balancing techniques for handling imbalanced flight delay datasets has received limited attention.

To address this problem, this study proposes a feature engineering framework for flight delay prediction by extracting three groups of engineered features namely Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs), and integrating them into the original flight delay dataset. In addition, to evaluate the effects of adding the newly extracted features to the original dataset, five widely used machine learning classifiers (i.e., Logistic Regression [5], Random Forest [6], TabNet [7], LightGBM [8] and XGBoost [9]) were trained and evaluated across five experimental scenarios. These scenarios included a baseline experiment using the original dataset, three experiments evaluating each engineered feature group (DDFs, ARSFs, and SIFs) individually, and a final experiment integrating all proposed feature groups. This experimental design enables a systematic assessment of both the individual predictive value and the complementary contribution of the engineered features. Furthermore, the dataset used in this study is imbalanced. To address the data balancing problem and reduce bias toward the majority class, two balancing techniques, namely Borderline-SMOTE [10] and SMOTE-Tomek Links (SMOTE-Tomek) [11], are employed. Finally, the performance of the classifiers is evaluated using several evaluation metrics such as accuracy, recall, specificity, precision, and area under the ROC curve (AUC).

Therefore, the main contributions of this study are summarized as follows:

- An integrated feature engineering framework is developed that complements the original flight dataset with Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs), providing additional operational, route-reliability, and schedule-intensity information for flight delay prediction.
- A comprehensive experimental evaluation is conducted using five machine learning models and two class-balancing techniques on a large real-world dataset, enabling a systematic assessment of the individual contribution of each proposed feature group as well as their combined effect on flight delay prediction performance.
- The study demonstrates that departure disruption, airline-route stability, and schedule intensity features provide complementary predictive information. Integrating all three feature groups with the original feature set yields superior prediction performance compared with combining the original feature set with each feature group individually.

The remainder of this study is organized as follows. Section II presents the related work on flight delay prediction. Section III describes the dataset and preprocessing procedures. Section IV presents the proposed feature engineering framework, the evaluated machine learning classifiers, and the experimental setup. Section V discusses the experimental results. Finally, Section VI presents the conclusions and future research directions.

II. Related Work

In recent decades, numerous statistical and probabilistic techniques have been applied for the purpose of predicting flight delays. Nevertheless, these traditional techniques tend to demonstrate poor performance when processing multidimensional datasets or capturing nonlinear relationships between variables [3]. For example, Tu et al. [12] introduced a statistical model to analyze and predict flight delays based on seasonal trends, daily propagation patterns, and random residuals. Additionally, Wong and Tsai [13] developed a statistical survival model based on Cox regression analysis to analyze delay propagation and variables that affect schedule reliability. However, considerable attention has recently been paid to developing methods for predicting flight delay occurrence to enhance Air Traffic Flow Management. Among classification approaches, several researchers have employed machine learning methods and achieved considerable predictive performance [3]. Accordingly, Natarajan et al. [14] applied two machine learning techniques, namely Logistic Regression and Decision Tree (DT). Their results showed that DT obtained the best results. Both models exhibited accuracy levels ranging from 80% to 85%. Moreover, Khaksar and Sheikholeslami [15] used DT, Bayesian modeling, RF, and a hybrid clustering-classification approach to predict flight delays. Their findings demonstrated that the hybrid approach achieved superior predictive performance in estimating flight delays. More recently, Alfarhood et al. [16] demonstrated that machine learning techniques can effectively predict flight delays using four models, namely CatBoost, XGBoost, LightGBM, and Multilayer Perceptron (MLP). Their experimental findings revealed that CatBoost achieved the highest accuracy and the lowest error rates. On the other hand, recent advances in deep learning methods have inspired researchers to investigate their ability to improve flight delay prediction and enhance the efficiency of air traffic data analysis [4]. Accordingly, Kim et al. [4] introduced two deep learning models for flight delay prediction, namely Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN). Their results showed that LSTM outperformed RNN in predictive performance. Furthermore, Ayaydın and Akcayol [17] applied Deep Recurrent Neural Networks (DRNNs), LSTM, and RF models to forecast flight delays, cancellations, and diversions. The authors demonstrated that the LSTM model achieved the best predictive performance, obtaining 96.50% recall. Moreover, Gui et al. [3] employed RF and Long Short-Term Memory (LSTM) models to predict flight delays and demonstrated that RF achieved the best predictive performance with 90.2% accuracy.

Feature engineering has become one of the key methods for improving flight delay predictions by creating useful variables based on flight data. Various researchers have proven that feature engineering is an effective method for capturing the operational and temporal patterns of flights. For instance, Jadhav and Kodavade [18] improved flight delay predictions by employing feature engineering to construct time, trend, and error features from raw flight data streams. They found that the inclusion of such features greatly enhanced the ability of several machine learning models to predict flight delays. Also, Afrane et al. [19] developed a flight delay prediction model that combined historical delay features with network centrality measures along with other weather and operations-related factors into a RF model. The proposed feature engineering approach achieved 92.36% accuracy. In the same year, Afrane et al. [20] introduced a feature engineering strategy that considers features such as airport connections and delay propagation. The results showed that the newly generated features enhanced the accuracy of predictions.

To address class imbalance and enhance prediction accuracy, many studies have employed balancing techniques. For example, Choi et al. [21] implemented DT, RF, AdaBoost, and k-Nearest Neighbors (KNN), and demonstrated that ensemble models combined with SMOTE and undersampling techniques significantly improved prediction accuracy for imbalanced flight delay datasets. Moreover, Yi et al. [22] proposed a Hybrid Stacking Ensemble model using KNN, RF, Logistic Regression, DT, and Gaussian Naive Bayes, combined with SMOTE and Boruta feature selection techniques. The authors found that the stacking model outperformed the individual classifiers, obtaining high performance scores above 0.80 across multiple evaluation metrics. Recent studies, such as AlBassam and AlShahrani [23], investigated the impact of three balancing techniques, namely Random Oversampling, SMOTE, and ADASYN, on flight delay prediction. Their results demonstrated that these techniques significantly improved predictive performance on imbalanced datasets. In particular, RF with Random Oversampling and DT with SMOTE achieved the best predictive accuracy of 90%.

Despite these advances, existing flight delay prediction studies have primarily focused on improving predictive algorithms or incorporating isolated categories of engineered features, including temporal, historical, network-based, and automatically extracted representations.

Although these approaches have demonstrated promising predictive performance, they often examine these feature categories independently. Consequently, limited attention has been given to developing an integrated feature engineering framework that simultaneously captures departure-stage operational disruptions, airline-route reliability, and schedule intensity characteristics. Furthermore, the interaction between such engineered features and class-balancing techniques for handling imbalanced flight delay datasets remains insufficiently explored. To address these limitations, this study proposes a unified feature engineering framework that integrates Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs). The effectiveness of the proposed feature groups is systematically evaluated using multiple machine learning classifiers and class-balancing techniques to assess both their individual and combined contributions to flight delay prediction.

III. Dataset Description and Preprocessing

The Flight Delay and Cancellation Dataset used in this study was downloaded from Kaggle[1]. The original data were sourced from the U.S. Department of Transportation, Bureau of Transportation Statistics. The dataset covers the period from 2019 to 2023 and contains 3,000,000 records with 32 features. To establish a realistic operational prediction setting, several post-arrival and leakage-prone features were dropped during the pre-processing phase. These included target features, operational features that were applicable after flight arrival, delay cause indicators after the event, and redundant features that were present for canceled or diverted flights. These variables were excluded because they directly give information on the target variable or become available only after the completion of the flight, or because they create data leakage in the prediction process. In addition, categorical features were encoded using one-hot encoding [24] to transform them into a numeric format suitable for machine learning models. After that, the remaining features were used to construct the final dataset for model development.

A. Class Label Construction and Data Balancing

For the classification process, the `ARR_DELAY` column was encoded as a binary target variable (0,1). A flight was classified as delayed when its arrival delay exceeded 15 minutes ($Delayed = 1$). On the other hand, a flight was classified as non-delayed when its arrival delay did not exceed 15 minutes ($Delayed = 0$). The 15-minute threshold is widely adopted in aviation studies because delays of this duration are considered operationally significant.

Furthermore, rows containing missing values in the `ARR_DELAY` column were removed since identifying the target class without actual arrival delay information was not possible. In addition, flights that were either canceled or diverted were excluded because they did not follow standard operational patterns. Following the preprocessing stage, class imbalance was observed, with 2,398,513 records belonging to the majority class (non-delayed flights) and 515,289 records belonging to the minority class (delayed flights). Accordingly, two balancing techniques were applied to address this issue:

- Borderline-SMOTE: This technique is an enhanced version of SMOTE that generates synthetic samples near the decision boundary between majority and minority classes. It primarily focuses on minority class samples with a high probability of misclassification [10].
- SMOTE with Tomek Links (SMOTE-Tomek): This hybrid technique combines SMOTE oversampling with Tomek Links undersampling. It generates synthetic minority class samples while simultaneously removing overlapping and noisy samples between classes [11].

IV. Methodology

This section describes the proposed flight delay prediction framework. The framework consists of three main stages: feature engineering, class imbalance handling, and predictive modeling. First, three groups of engineered features (i.e., Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs)) were extracted to capture operational disruption, airline-route stability, and schedule intensity characteristics and integrated with the original flight delay dataset. Then, two balancing techniques (i.e., Borderline-SMOTE and SMOTE-Tomek) were applied to address the class imbalance problem. Finally, various machine learning classifiers (i.e., LightGBM, TabNet, RF, Logistic Regression, and XGBoost) were evaluated to assess the effectiveness of the proposed feature engineering framework. To systematically evaluate the contribution of the engineered features, five experimental scenarios were considered, including a baseline experiment using the original feature set, three experiments combining the original feature set separately with DDFs, ARSFs, and SIFs, and a final experiment combining the original feature set with all three engineered feature groups. Fig. 1 illustrates the workflow of the proposed feature engineering framework.

A. Design of the Proposed Engineered Features

In this subsection, three groups of engineered features were extracted and integrated with the original flight delay dataset to improve flight delay prediction performance. These features were specifically designed to address operational disruption, airline-route reliability, and schedule characteristics, which are not directly represented by existing variables. Features are classified into the following three groups: Departure Disruption Features (DDFs), Airline Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs). To avoid data leakage, all engineered features were constructed using operational, scheduled, and historical information available before arrival and without using any arrival-stage variables or target-related information.

[1]https://www.kaggle.com/datasets/patrickzel/flight-delay-and-cancellation-dataset-2019-2023

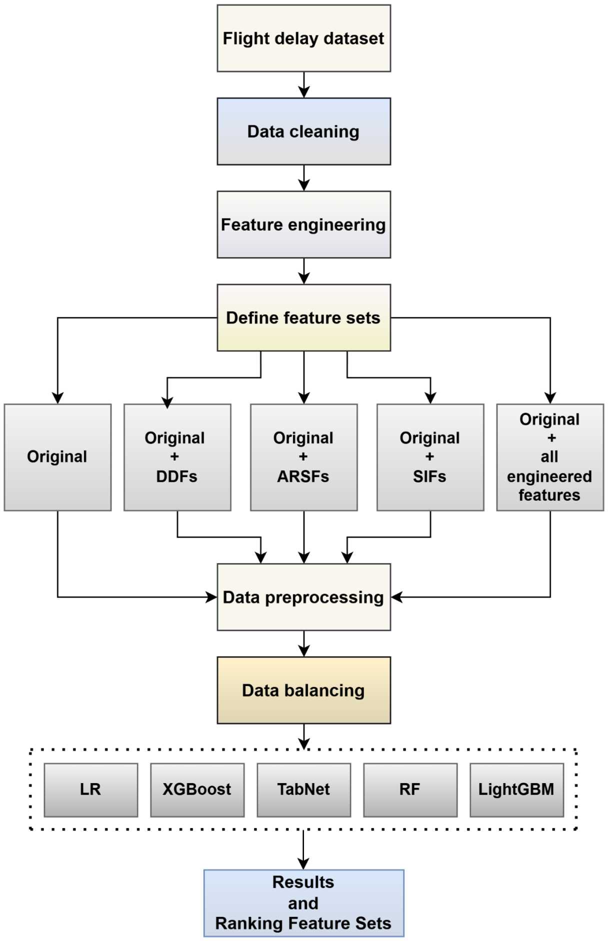

Fig. 1. Proposed flight delay prediction framework.

1) Departure Disruption Features (DDFs): Departure disruptions are one of the most influential factors affecting on-time performance. While the original data includes features such as departure delay and taxi-out time these variables do not explicitly measure operational stress that may happen before take-off. For this reason, four Departure Disruption Features (DDFs) were extracted to capture several departure stage disruptions.

a) Positive Departure Delay (PDD): This feature refers to the amount of departure delay that does not consider early departures. This is determined by:

$$PDD_i = \max(\text{DEP\_DELAY}_i, 0) \tag{1}$$

where, DEP_DELAY_i, refers to the departure delay of flight i. Although DEP_DELAY is available in the original dataset, negative values represent early departures, which do not necessarily indicate reduced delay risk. Therefore, Positive Departure Delay isolates the operational burden associated with late departures by retaining only positive delay values. Since the risk of arrival delay is primarily linked to late departures rather than early departures, this feature provides a more meaningful representation of departure-stage operational disruption before take-off.

b) Departure Operational Pressure (DOP): This feature normalizes departure-related operational burden by scheduled flight duration, enabling meaningful comparison between short-haul and long-haul flights. The same delay burden may have a greater operational impact on shorter flights than on longer flights. This is expressed as:

$$DOP_i = \frac{PDD_i + \text{TAXI\_OUT}_i}{\text{CRS\_ELAPSED\_TIME}_i + \varepsilon} \tag{2}$$

where, DOP_i denotes the Departure Operational Pressure of flight i, $TAXI\_OUT_i$ denotes the taxi-out time, and $CRS\_ELAPSED\_TIME_i$ represent the scheduled flight duration. Furthermore, ε is a small positive constant added to avoid division-by-zero errors. This attribute provides a normalized metric of departure-stage operational burden in flights of different intervals.

c) Taxi-Out Congestion Burden (TOCB): Taxi-Out Congestion Burden defines airport surface congestion through measuring taxi-out operation against the schedule duration of the flight.

$$TOCB_i = \frac{\text{TAXI\_OUT}_i}{\text{CRS\_ELAPSED\_TIME}_i + \varepsilon} \tag{3}$$

where, $TOCB_i$ denotes the Taxi-Out Congestion Burden of flight i. A higher value implies that a larger proportion of the planned journey time is consumed before takeoff, suggesting higher congestion levels.

d) Severe Departure Delay Indicator (SDDI): To record extremely disruptive departure delays, a feature termed SDDI was created:

$$SDDI_i = \begin{cases} 1, & \text{if } PDD_i \geq 60, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This feature identifies flights experiencing severe operational disruption (departure delay $\geq$ 60 minutes). Such extreme delays may exhibit nonlinear effects on arrival performance compared with moderate delays. All together, these variables constitute a complete description of potential departure stage disruption, airport congestion, and schedule recovery restrictions which can lead to arrival delay.

2) Airline-Route Stability Features (ARSFs): Flight delays display distinct patterns with respect to routes and airlines due to factors such as operational efficiency, airport conditions, network topology, and scheduling practices. The following three Airline-Route Stability Features (ARSFs) were designed to capture these historical reliability characteristics. Departure delay statistics were selected because they provide route-level operational reliability information while avoiding direct dependence on the arrival-delay target variable.

a) Historical Route Delay Mean (HRDM): It is the mean value of historical departure delay of a specific airline-route pair and can be determined by:

$$HRDM_r = \frac{\sum_{j \in R_r} \text{DEP\_DELAY}_j}{n_r} \tag{5}$$

where, R_r represents the set of historical flights belonging to airline-route r, and n_r is the number of flights within that route group. It represents the long-term departure delay tendency in a certain route for a specific airline.

b) Historical Route Delay Variability (HRDV): While the mean delay shows route disposition, it fails to show route stability. As a result, Historical Route Delay Variability has been devised to determine the consistency of the route and calculated as:

$$HRDV_r = \sqrt{\frac{\sum_{j \in R_r} \left(\text{DEP\_DELAY}_j - HRDM_r\right)^2}{n_r - 1}} \tag{6}$$

A higher value indicates greater operational instability and less predictable route behavior. Routes with high historical variability exhibit less predictable operational behavior and may therefore be more difficult to classify accurately than routes with consistent delay patterns.

c) Route Historical Sample Size (RHSS): This feature represents the amount of historical data that can be accessed for a particular route-airline combination:

$$RHSS_r = n_r \tag{7}$$

This feature helps distinguish well-established routes with abundant historical observations from routes with limited operational history. In addition, it provides an indication of the reliability of route-level historical statistics, since estimates derived from larger numbers of flights are generally more stable than those derived from sparse observations. In summary, the ARSFs group incorporates both the expected delay pattern and the operational stability of airline-route combinations, providing the prediction models with valuable historical reliability data. To prevent data leakage, the historical statistics used to construct the ARSFs were estimated independently within each cross-validation fold. For every fold, HRDM, HRDV, and RHSS were computed exclusively from the fold-training data and mapped to the corresponding validation fold without recomputation. For the final holdout evaluation, these statistics were estimated again using the complete development set and applied to the test set.

3) Schedule Intensity Features (SIFs): The Schedule design influences the ability of flights to absorb operational disruptions. Those running on an intense schedule are less likely to recover delays. Therefore, three Schedule Intensity Features (SIFs) were derived to measure scheduling attributes that may cause delays.

a) Scheduled Flight Speed (SFS): This feature measures the planned travel speed of a flight and is expressed as:

$$SFS_i = \frac{\text{DISTANCE}_i}{\left(\frac{\text{CRS\_ELAPSED\_TIME}_i}{60}\right) + \varepsilon} \tag{8}$$

where, $DISTANCE_i$ represents the flight distance and $CRS\_ELAPSED\_TIME_i$ denotes the scheduled duration. Higher scheduled speeds may indicate more aggressive scheduling practices and reduced opportunities for delay recovery. Flights operating under tighter schedules may have fewer opportunities to absorb operational disruptions and recover from delays.

b) Departure Time Cyclicity$_{sin}$ (DTC$_{sin}$): Departure time is inherently cyclical because the start and end of the day are close in terms of time. Direct numerical encoding of departure time fails to preserve the circular nature of clock time, since 23:59 and 00:01 are temporally close despite being numerically distant. To retain the cyclical nature of departure time and avoid this artificial discontinuity, the scheduled departure time was modeled as follows:

$$DTC_{\sin} = \sin\left(\frac{2\pi \times \text{SDM}_i}{1440}\right) \tag{9}$$

where, SDM_i represents the scheduled departure time expressed as minutes after midnight and 1440 corresponds to the total number of minutes in a day.

c) Departure Time Cyclicity$_{cos}$ (DTC$_{cos}$): To complement the sine representation and uniquely identify time-of-day positions, a cosine transformation is performed as follows:

$$DTC_{\cos} = \cos\left(\frac{2\pi \times \text{SDM}_i}{1440}\right) \tag{10}$$

The combination of $sine$ and $cosine$ components preserves temporal continuity and enables the learning algorithms to model daily traffic cycles, peak times, and scheduling patterns without generating any artificial break points at midnight. Overall, the SIF group consists of schedule tightness and time-dependent characteristics that may affect flight delays based only on pre-departure data.

Together, the proposed engineered features capture three complementary dimensions of flight operations. DDFs characterize departure-stage disruptions and congestion effects, ARSFs represent long-term airline-route reliability and stability patterns, and SIFs describe schedule design and temporal operating characteristics. The proposed engineered features transform and combine available operational, historical, and scheduling variables into representations that more explicitly characterize flight delay related conditions. PDD distinguishes late departures from early departures, DOP and TOCB represent normalized departure-stage operational conditions, while SFS and the cyclic time features characterize schedule intensity and temporal patterns. The experimental scenarios therefore evaluate whether these engineered representations provide additional predictive value beyond using the original variables directly.

B. Classifiers Employed

To evaluate the effectiveness of the proposed engineered features under different feature-set configurations, five well-known machine learning classifiers were employed. These classifiers were selected because they represent diverse learning procedures, including linear models, ensemble tree methods, gradient boosting algorithms, and deep learning architectures. Moreover, evaluating multiple classifiers enables a thorough analysis of their predictive performance and a robust evaluation of the impact of the introduced features on flight delay prediction. Accordingly, the classifiers used in this study are:

1) Logistic Regression (LR): it is a probabilistic linear classifier, where the probability of the positive class is obtained by applying sigmoid function to a linear combination of input features. Parameters are estimated using the maximum likelihood estimation, which is essentially the minimization of the binary cross-entropy loss [5].

2) Random Forest (RF): it is an ensemble machine learning classifier that integrates multiple decision trees using bagging and majority voting techniques. Moreover, RF can effectively reduce overfitting, handle high-dimensional datasets, and capture diverse operational patterns within the data [6], [25].

3) XGBoost: it is an ensemble learning method that sequentially builds decision trees, where each new tree focuses on correcting the mistakes made by the previous ones, gradually improving overall prediction accuracy. It uses gradient boosting under the hood, combined with built-in regularization techniques to keep the model from overfitting. Its speed and performance on structured/tabular data have made it one of the most popular and reliable algorithms in machine learning competitions and real-world applications [9].

4) LightGBM: It is an advanced gradient boosting decision tree classifier. LightGBM constructs decision trees sequentially in order to minimize prediction errors generated by previous trees. Furthermore, LightGBM is highly effective when handling large datasets and modeling complex nonlinear relationships [8], [26].

5) TabNet: It is a deep learning model developed specifically for handling tabular datasets. it utilizes sequential attention technique to determine and focus on the most informative features during each decision stage. TabNet provides an interpretable and efficient learning framework by integrating feature selection with nonlinear representation learning within a unified architecture [7].

6) Experimental setup: After preprocessing, the dataset was divided using stratified sampling into an 80% development set and an independent 20% holdout test set, preserving the delayed and non-delayed class proportions. Stratified 10-fold cross-validation was conducted exclusively on the development set for model training, comparison, and selection. In each fold, nine folds were used for training and the remaining fold was used for validation.

To prevent information leakage, all data-dependent preprocessing operations were fitted independently within each cross-validation fold. Specifically, the ARSFs (HRDM, HRDV, and RHSS) were estimated exclusively from the nine training folds and subsequently mapped to the corresponding validation fold without recomputation. Similarly, categorical encoding and numerical normalization were fitted using only the fold-training data. Borderline-SMOTE and SMOTE-Tomek were also applied exclusively to the fold-training portion, while the validation fold remained unchanged.

Performance metrics were computed for each validation fold and averaged across the ten folds. After model and feature-set selection, the selected configuration was retrained on the complete 80% development set. The ARSF statistics and preprocessing transformations were re-estimated using only this development set, and final performance was evaluated once on the untouched 20% holdout test set.

V. Results

This section discusses and analyzes the results obtained from five different scenarios that were conducted to evaluate the effectiveness of the proposed feature engineering method. The first experiment evaluates the performance of the classifiers using the original dataset. The second, third, and fourth experiments investigate the impact of incorporating the proposed DDFs, ARSFs, and SIFs. Finally, the fifth experiment evaluates the effects of adding all the proposed features together with the original dataset. The comparison was conducted among several different classifiers and balancing techniques to evaluate the effectiveness of the generated features on enhancing the prediction accuracy.

A. Evaluation Metrics

Since class imbalance may result in biased outcomes of the majority class, accuracy alone is not sufficient to evaluate the performance of the classifiers. Therefore, for a more comprehensive assessment, four other metrics were used. These metrics were calculated using the confusion matrix to evaluate the prediction performance of the models. Table I illustrates the confusion matrix, where TP and TN denote correctly classified delayed and non-delayed flights, respectively, while FP and FN denote non-delayed flights incorrectly classified as delayed and delayed flights incorrectly classified as non-delayed, respectively. [22], [27].

TABLE I. Confusion Matrix

		Predicted class	
		Delay	**Non delay**
Actual class	**Delay**	**TP**	**FN**
	Non delay	**FP**	**TN**

- **Accuracy**: Measures the proportion of correctly classified flights from the total number of flights.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

- **Precision**: Measures the proportion of flights predicted as delayed that were actually delayed.

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

- **Recall**: Measures the ability of the model to correctly identify actual delayed flights.

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

- **Specificity**: Measures the model's capability to accurately classify non-delayed flights.

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

TABLE II. CLASSIFICATION PERFORMANCE OF THE EVALUATED MODELS USING THE ORIGINAL DATASET WITH DIFFERENT DATA BALANCING TECHNIQUES

Model	Accuracy	Recall	Specificity	Precision	AUC
Imbalanced Dataset					
Logistic Regression	0.9286	**0.8970**	0.9354	0.7489	0.9660
RF	0.9449	0.7414	**0.9886**	**0.9334**	0.9538
XGBoost	0.9506	0.7935	0.9843	0.9159	0.9651
LightGBM	**0.9530**	0.8046	0.9848	0.9193	**0.9674**
TabNet	0.9499	0.7849	0.9854	0.9201	0.9647
Borderline-SMOTE					
Logistic Regression	0.9248	0.9014	0.9298	0.7341	0.9660
RF	0.9209	0.8843	0.9288	0.7273	0.9549
XGBoost	0.9304	0.8865	0.9398	0.7598	0.9643
LightGBM	**0.9473**	0.8461	**0.9690**	**0.8545**	**0.9662**
TabNet	0.9162	**0.9025**	0.9191	0.7057	0.9645
SMOTE-Tomek					
Logistic Regression	0.9316	0.8922	0.9400	0.7616	0.9659
RF	0.9285	0.8754	0.9399	0.7578	0.9558
XGBoost	0.9363	0.8738	0.9497	0.7887	0.9643
LightGBM	**0.9487**	0.8380	**0.9725**	**0.8675**	0.9663
TabNet	0.9216	**0.9043**	0.9253	0.7224	**0.9673**

- **AUC**: Measures the model's ability to differentiate between delayed and non-delayed flights.

B. Results Discussion

This subsection presents and analyzes the performance of the proposed feature engineering framework for flight delay prediction. Five machine learning classifiers were evaluated in five feature engineering scenarios, including the original feature set, the original feature set combined individually with the proposed Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs), as well as the combination of all proposed feature groups. To address the data balancing problem, the experiments were conducted using the original imbalanced dataset, and with Borderline-SMOTE and SMOTE-Tomek balancing techniques. The classifiers were evaluated using five performance metrics: accuracy, recall, specificity, precision, and AUC.

1) Results obtained using the original dataset: In the first experiment, five classifiers namely Logistic Regression, RF, XGBoost, LightGBM and TabNet were evaluated using the original flight delay dataset. To address class balancing problem, Borderline-SMOTE and SMOTE-Tomek were used. The performance of the classifiers was evaluated using accuracy, precision, recall, specificity and AUC.

As shown in Table II LightGBM achieved the highest accuracy and AUC, demonstrating excellent discrimination between delayed and non-delayed flights. Logistic Regression had the highest recall, which demonstrated the efficiency of the model in detecting delayed flights. In contrast, RF delivered the highest specificity and precision but the lowest recall, reflecting bias towards the majority class (non-delayed flights). Applying the two balancing techniques generally improved recall in most classifiers, showing enhanced detection of delayed flights. However, these improvements were associated with a decrease in precision and specifically, reflecting a trade-off between identifying more delayed flights. SMOTE-Tomek produced balanced results by maintaining relatively high recall with better accuracy and specificity. Overall, LightGBM showed the most balanced performance in both sampling techniques. SMOTE-Tomek generally provided a more favorable trade-off between recall and overall classification performance than Borderline-SMOTE.

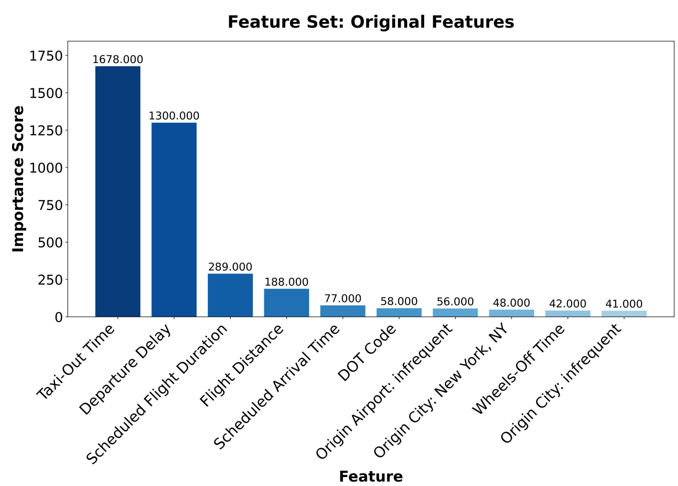

Fig. 2. Ranking of the ten most important features identified by LightGBM applied on the Original feature set after SMOTE-Tomek balancing.

Fig. 2 shows the ranking of features based on the importance of each of them in the original set of features for the combination of LightGBM and SMOTE-Tomek (Best performer). Taxi-out time and departure delay were the most important factors, indicating their strong predictive relevance to flight delays. In addition, the scheduled duration and distance of the

TABLE III. CLASSIFICATION PERFORMANCE OF THE EVALUATED MODELS USING THE ORIGINAL + DDFS FEATURE SET WITH DIFFERENT DATA BALANCING TECHNIQUES

Model	Accuracy	Recall	Specificity	Precision	AUC
Imbalanced Dataset					
Logistic Regression	0.9405	**0.9062**	0.9478	0.7671	0.9765
RF	0.9604	0.7918	0.9961	0.9386	0.9746
XGBoost	0.9620	0.8077	0.9968	0.9317	0.9772
LightGBM	**0.9637**	0.8150	**0.9972**	0.9348	**0.9794**
TabNet	0.9591	0.7855	0.9960	**0.9391**	0.9728
Borderline-SMOTE					
Logistic Regression	0.9268	0.9206	0.9281	0.7154	0.9768
RF	0.9330	0.9082	0.9384	0.7392	0.9736
XGBoost	0.9342	0.9090	0.9397	0.7425	0.9748
LightGBM	**0.9562**	0.8625	**0.9765**	**0.8601**	**0.9779**
TabNet	0.9114	**0.9272**	0.9081	0.6670	0.9753
SMOTE-Tomek					
Logistic Regression	0.9432	0.9021	0.9520	0.7802	0.9766
RF	0.9486	0.8830	0.9628	0.8116	0.9749
XGBoost	0.9491	0.8840	0.9632	0.8130	0.9755
LightGBM	**0.9600**	0.8460	**0.9865**	**0.8884**	**0.9780**
TabNet	0.8820	**0.9380**	0.8700	0.5960	0.9710

flights were also significant features.

2) Results obtained using original features + DDFs: In the second experiment, the proposed Departure Disruption Features (DDFs) were extracted and incorporated into the original flight delay dataset. These features were designed to obtain disruption-related factors that may influence flight reliability. Additionally, Borderline-SMOTE and SMOTE-Tomek were employed to solve class imbalance. The performance of the models was assessed using five metrics to determine whether the inclusion of DDFs improves overall performance.

According to the results in Table III, incorporating Departure Disruption Features (DDFs) into the original dataset improved the performance of all classifiers compared to the first experiment. In the imbalanced dataset, all classifiers showed greater values in terms of accuracy, AUC, precision, and specificity compared to those achieved using the original feature set, which shows the efficiency of DDFs for describing disruption-relevant information related to flight delay. LightGBM achieved the highest accuracy, AUC and specificity. After applying Borderline-SMOTE and SMOTE-Tomek recall values improved in most classifiers, indicating greater detection of (minority class) delayed flights. TabNet + SMOTE-Tomek produced the best recall, followed by TabNet + Borderline-SMOTE. However, there was a reduction in the precision and specificity measures, reflecting the trade-off between identifying more delayed flights. SMOTE-Tomek generally produced more balanced results. LightGBM combined with SMOTE-Tomek demonstrated the strongest and most consistent performance.

As shown in Fig. 3, three of the proposed DDFs (i.e., Positive Departure Delay, Departure Operational Pressure, and Taxi-Out Congestion Burden), are ranked among the ten most important features. It shows that the engineered features contribute substantially to predictive value and are actively applied by the classifier during decision making. The high importance assigned to these features suggests that they provide useful disruption-related information beyond the original feature representation. As a result, the improved classification performance reported in Table III is consistent with the ability of the proposed DDFs to provide a more informative representation of operational conditions associated with flight delays.

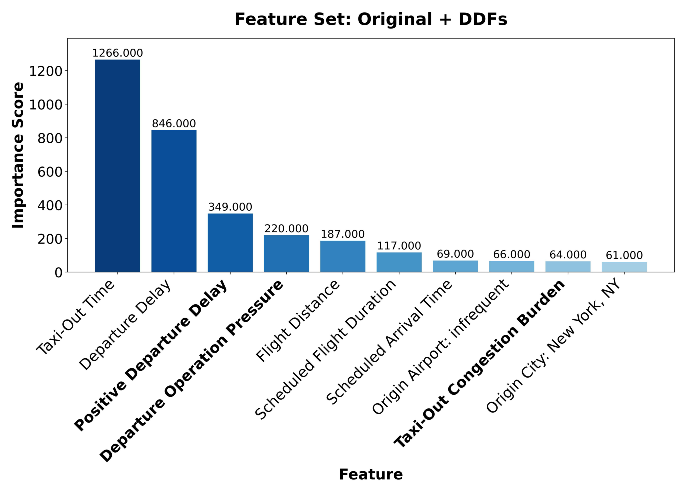

Fig. 3. Ranking of the ten most important features identified by LightGBM applied on the Original + DDFs feature set after SMOTE-Tomek balancing.

3) Results obtained using original features + ARSFs: In the third experiment, Airline-Route Stability (ARSFs) were incorporated into the original flight delay dataset. The ARSFs were created to capture the stability and reliability properties of the airlines as well as flight routes. Two balancing techniques were employed to address class balancing problem. The same evaluation metrics were used to evaluate the effectiveness of

TABLE IV. CLASSIFICATION PERFORMANCE OF THE EVALUATED MODELS USING THE ORIGINAL + ARSFS FEATURE SET WITH DIFFERENT DATA BALANCING TECHNIQUES

Model	Accuracy	Recall	Specificity	Precision	AUC
Imbalanced Dataset					
Logistic Regression	0.9418	**0.9112**	0.9483	0.7635	0.9774
RF	0.9521	0.6924	**0.9990**	**0.9508**	0.9628
XGBoost	0.9647	0.8081	0.9964	0.9291	0.9776
LightGBM	**0.9670**	0.8194	0.9969	0.9338	**0.9801**
TabNet	0.9498	0.7287	0.9968	0.9240	0.9465
Borderline-SMOTE					
Logistic Regression	0.9406	**0.9121**	0.9467	0.7587	0.9770
RF	0.9318	0.8462	0.9505	0.7534	0.9598
XGBoost	0.9462	0.8986	0.9567	0.7818	0.9765
LightGBM	**0.9615**	0.8588	**0.9851**	**0.8768**	**0.9785**
TabNet	0.9394	0.9093	0.9462	0.7561	0.9780
SMOTE-Tomek					
Logistic Regression	0.9457	**0.9062**	0.9548	0.7797	0.9768
RF	0.9384	0.8527	0.9561	0.7776	0.9624
XGBoost	0.9510	0.8862	0.9664	0.8116	0.9762
LightGBM	**0.9620**	0.8510	**0.9886**	**0.8879**	**0.9787**
TabNet	0.9430	0.8967	0.9529	0.7774	0.9757

the developed ARSFs features.

The results presented in Table IV demonstrate that the incorporation of the ARSFs features improved the performance of the classifiers. Using the imbalanced dataset, LightGBM had the highest accuracy and AUC, while maintaining high specificity and precision. Conversely, Logistic Regression provided the best recall, implying its capability to predict delayed flights. RF obtained the top specificity and precision but low recall which indicated that they tend to favor the majority class. The balancing techniques: Borderline-SMOTE and SMOTE-Tomek substantially enhanced the ability of the classifiers to identify delayed flights. These improvements were especially noticeable in the Logistic Regression and TabNet models. However, there was a slight decline in precision and specificity, meaning that the models became more prone to predict the delayed class. Compared to Borderline-SMOTE, SMOTE-Tomek demonstrated a better balance between sensitivity towards the minority class and overall classification accuracy. Among all experiments, LightGBM performed better compared to other classifiers regarding accuracy and AUC.

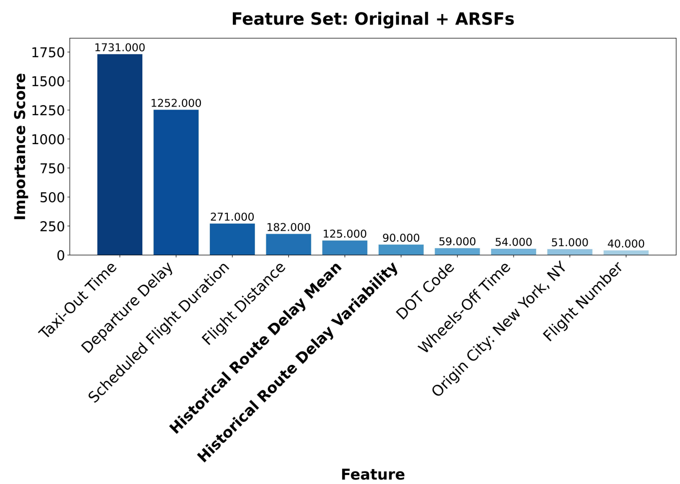

Fig. 4. Ranking of the ten most important features identified by LightGBM applied on the Original + ARSFs feature set after SMOTE-Tomek balancing.

Fig. 4 demonstrates that two of the proposed ARSFs, namely *Historical Route Delay Mean* and *Historical Route Delay Variability*, are ranked among the ten most important features. This indicates that the engineered features provide substantial predictive value and are effectively exploited by the classifier. The importance of the attributes shows that the features represent consistent delay behavior and stability at the route level which cannot be found in the initial attributes set. In other words, the superior classification results shown in Table IV are because the suggested ARSFs can extend the feature set with the relevant historical airline route reliability data for flight delay prediction.

4) Results obtained using original features + SIFs: In the fourth experiment, additional features, known as the Schedule Intensity Indicators (SIFs), were added into the existing flight delay dataset to capture schedule-based or congestion-related variables. The same experimental setup used previously, Borderline-SMOTE, SMOTE-Tomek were employed to solve class imbalance, and the same evaluation metrics were used.

As shown in Table V the proposed Schedule Intensity Features (SIFs) enhanced the performance of the classifiers compared with the original dataset. LightGBM consistently achieved the highest accuracy and AUC. while Logistic Regression achieved the highest recall. Meanwhile, RF had the best specificity and precision but lower recall. Both Borderline-SMOTE and SMOTE-Tomek improved the ability of the classifiers to detect delayed flights. Nevertheless, this higher

TABLE V. CLASSIFICATION PERFORMANCE OF THE EVALUATED MODELS USING THE ORIGINAL + SIFS FEATURE SET WITH DIFFERENT DATA BALANCING TECHNIQUES

Model	Accuracy	Recall	Specificity	Precision	AUC
Imbalanced Dataset					
Logistic Regression	0.9426	**0.9124**	0.9491	0.7668	0.9781
RF	0.9591	0.7589	**0.9981**	**0.9395**	0.9705
XGBoost	0.9660	0.8145	0.9970	0.9316	0.9784
LightGBM	**0.9682**	0.8233	0.9974	0.9367	**0.9808**
TabNet	0.9658	0.8169	0.9966	0.9290	0.9780
Borderline-SMOTE					
Logistic Regression	0.9414	0.9152	0.9470	0.7608	0.9778
RF	0.9402	0.8906	0.9509	0.7642	0.9709
XGBoost	0.9483	0.8987	0.9590	0.7868	0.9770
LightGBM	**0.9625**	0.8609	**0.9858**	**0.8780**	**0.9794**
TabNet	0.9148	**0.9340**	0.9113	0.6685	0.9768
SMOTE-Tomek					
Logistic Regression	0.9465	0.9071	0.9550	0.7828	0.9776
RF	0.9445	0.8881	0.9565	0.7829	0.9720
XGBoost	0.9524	0.8849	0.9665	0.8148	0.9769
LightGBM	**0.9633**	0.8524	**0.9888**	**0.8887**	**0.9792**
TabNet	0.9354	**0.9190**	0.9392	0.7336	0.9785

sensitivity was accompanied by reduced precision and specificity, reflecting an increased probability of classifying delays. Overall, SMOTE-Tomek provided a favorable balance across the evaluated metrics, although its effect varied across classifiers. LightGBM consistently emerged as the most effective and balanced performer in comparison to other methods.

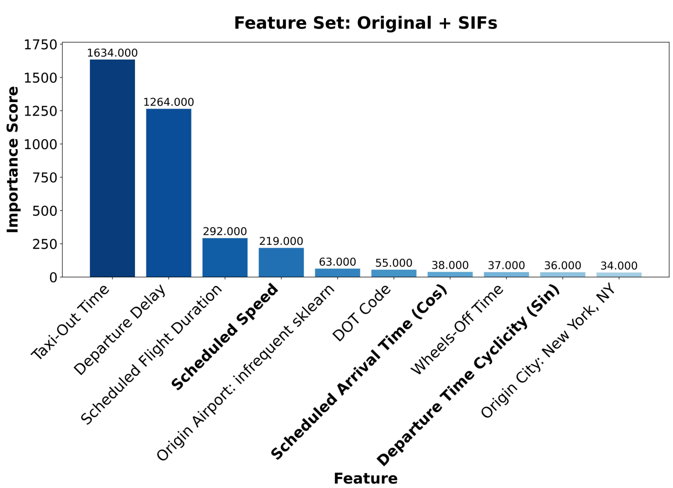

Fig. 5. Ranking of the ten most important features identified by LightGBM applied on the Original + SIFs feature set after SMOTE-Tomek balancing.

In Fig. 5, it can be observed that multiple SIFs, such as Scheduled Speed and time-related scheduling variables, are among the ten most important features for the combination of LightGBM and SMOTE-Tomek. This indicates that schedule intensity and temporal traffic variables provide relevant predictive information for flight punctuality. As a result, enhanced classification performance can contribute to the ability of the proposed SIFs to provide a more informative representation of scheduling characteristics associated with flight delays.

5) Results obtained using original features + DDFs + ARSFs + SIFs: In the fifth experiment, all proposed features (i.e., DDFs, ARSFs, and SIFs) were integrated with the original flight delay dataset. The aim of this step is to analyze the cumulative effect of these engineered features and determine whether these features provide complementary information. As with the previous experiments, Borderline-SMOTE and SMOTE-Tomek techniques were adopted to handle class imbalance, and the same evaluation metrics were used to assess classification performance.

Table VI reveals that combining DDFs, ARSFs and SIFs yielded the best performance in all experiments. In the original dataset LightGBM achieved the highest values in several metrics such as accuracy, AUC and specificity. Logistic Regression obtained the highest recall. All classifiers showed substantial improvement in terms of accuracy, precision, specificity, and AUC compared with the previous experiments, indicating that the newly proposed feature groups have complementary predictive value when used together. Both Borderline-SMOTE and SMOTE-Tomek increased the capability of the classifiers to detect delayed flights, hence an increase in recall value. Logistic Regression combined with Borderline-SMOTE achieved the highest recall. However, this improvement was associated with decreased precision and specificity, showing the trade-off between identifying more delayed flights. Among the balancing techniques, SMOTE-Tomek consistently provided more balanced outcomes. LightGBM combined with SMOTE-Tomek delivered the strongest performance.

Fig. 6 displays the order of feature importance scores obtained from LightGBM + SMOTE-Tomek applied to the combined feature sets. Besides the baseline features, several engineered features derived from DDFs, ARSFs and SIFs were found to be among the most significant features including *Positive Departure Delay*, *Departure Operational Pressure*, *Scheduled Speed*, *Taxi-Out Congestion Burden*, and *Departure*

TABLE VI. CLASSIFICATION PERFORMANCE OF THE EVALUATED MODELS USING THE ORIGINAL + DDF$_s$ + ARSF$_s$ + SIF$_s$ FEATURE SET WITH DIFFERENT DATA BALANCING TECHNIQUES

Model	Accuracy	Recall	Specificity	Precision	AUC
Imbalanced Dataset					
Logistic Regression	0.9612	**0.9298**	0.9667	0.7928	0.9879
RF	0.9791	0.8204	0.9992	**0.9582**	0.9854
XGBoost	0.9813	0.8358	0.9994	0.9545	0.9871
LightGBM	**0.9838**	0.8406	**0.9995**	0.9571	**0.9888**
TabNet	0.9785	0.8245	0.9990	0.9500	0.9835
Borderline-SMOTE					
Logistic Regression	0.9548	**0.9388**	0.9566	0.7615	0.9867
RF	0.9606	0.9229	0.9674	0.7885	0.9838
XGBoost	0.9635	0.9255	0.9708	0.7976	0.9852
LightGBM	**0.9777**	0.8839	**0.9942**	**0.8916**	**0.9880**
TabNet	0.9505	0.9364	0.9507	0.7394	0.9850
SMOTE-Tomek					
Logistic Regression	0.9654	0.9236	0.9726	0.8079	0.9869
RF	0.9709	0.9048	0.9818	0.8361	0.9840
XGBoost	0.9731	0.9035	0.9841	0.8432	0.9850
LightGBM	**0.9820**	0.8721	**0.9960**	**0.9137**	**0.9884**
TabNet	0.9462	**0.9368**	0.9468	0.7275	0.9839

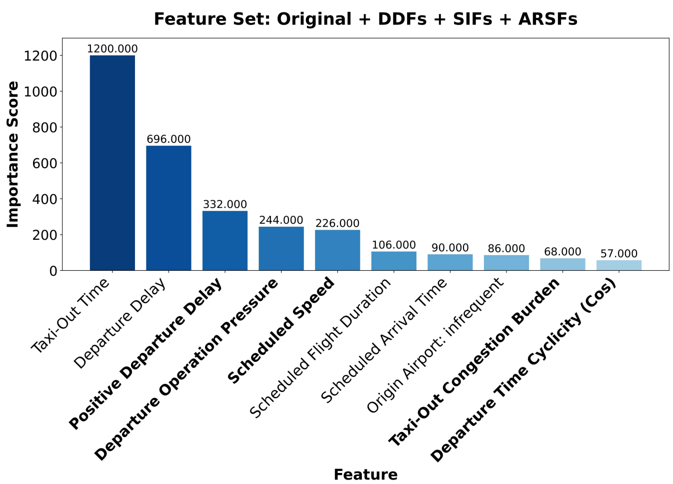

Fig. 6. Ranking of the ten most important features identified by LightGBM applied on the Original + DDF + SIFs + ARSFs feature set after SMOTE-Tomek balancing.

Time Cyclicity (Cos). The inclusion of important features in all categories suggests that all categories complement each other by providing predictive information. This can explain the superior classification performance provided by the combined set of features.

Overall, the feature importance analysis reveals the success of the proposed feature engineering approach. While the impact of operational features, including *Taxi-Out Time* and *Departure Delay*, remained substantial for all experiments, the engineered features of DDFs, ARSFs and SIFs were always rated among the top important features. This is why the positive effect on the classification accuracy can be explained by their combination that covers various aspects of the problem.

The effect of class balancing varied across classifiers and feature sets. Although both balancing techniques generally improved recall, this improvement was often accompanied by lower precision and specificity, reflecting a trade-off between detecting delayed flights and maintaining overall classification performance. SMOTE-Tomek generally provided a better balance than Borderline-SMOTE, particularly with LightGBM, but this pattern was not consistent across all classifiers. For example, with TabNet, SMOTE-Tomek increased recall in some scenarios while reducing accuracy and precision. These results indicate that the effectiveness of the balancing technique depends on both the classifier and the feature-set configuration.

C. Summary of Results

In summary, Table VII demonstrates the effectiveness of the proposed feature engineering framework for improving flight delay prediction performance. Compared with the baseline feature set, the inclusion of DDFs, ARSFs, and SIFs individually improved the predictive performance across most evaluation metrics. The individual feature groups produced comparable performance improvements over the baseline, while combining all three feature groups yielded the highest overall performance. This indicates that they provide complementary predictive information. Across all experiments, LightGBM consistently delivered the strongest overall performance in terms of accuracy, AUC, precision, and specificity, whereas Logistic Regression achieved the highest recall. Furthermore, SMOTE-Tomek generally outperformed Borderline-SMOTE by providing a better balance between minority-class detection and overall classification performance. The combination of LightGBM, SMOTE-Tomek, and the complete feature set (DDFs + ARSFs + SIFs) achieved the best overall balanced results, demonstrating the value of integrating operational disruption, airline-route stability, and schedule intensity char-

TABLE VII. Summary of Classification Performance Across All Feature Engineering Scenarios Using the Best-Performing Model

Feature Set	Model	Accuracy	Recall	Specificity	Precision	AUC
Original	LightGBM + SMOTE-Tomek	0.9487	0.8380	0.9725	0.8675	0.9663
Original + DDFs	LightGBM + SMOTE-Tomek	0.9600	0.8460	0.9865	0.8884	0.9780
Original + ARSFs	LightGBM + SMOTE-Tomek	0.9620	0.8510	0.9886	0.8879	0.9787
Original + SIFs	LightGBM + SMOTE-Tomek	0.9633	0.8524	0.9888	0.8887	0.9792
Original + DDFs + ARSFs + SIFs	LightGBM + SMOTE-Tomek	**0.9820**	**0.8721**	**0.9960**	**0.9137**	**0.9884**

acteristics for flight delay prediction.

VI. Conclusion and Future Work

In this study, a feature engineering framework was developed to address the problem of flight delay prediction using three groups of engineered features, namely Departure Disruption Features (DDFs), Airline-Route Stability Features (ARSFs), and Schedule Intensity Features (SIFs), each group comprising a set of features designed to capture operational disruptions, route reliability, and scheduling characteristics. These features were integrated into the dataset. To address the class imbalance problem, two balancing techniques, namely Borderline-SMOTE and SMOTE-Tomek, were employed. Five classification models were employed across five experimental scenarios, including a baseline configuration, individual evaluations of DDFs, ARSFs, and SIFs, and a final integrated configuration, to assess the effectiveness of the proposed engineered features.

The results demonstrated that the engineered features improved predictive performance by capturing additional operational, historical, and scheduling characteristics that were not explicitly represented in the original dataset. More importantly, the performance of the integrated feature set indicates that departure disruption, airline-route stability, and schedule intensity provide complementary information and are more informative when considered jointly. The findings also show that the effect of class balancing varies across classifiers and feature configurations, highlighting the importance of selecting a balancing strategy according to the predictive model and the desired trade-off between delayed-flight detection and overall classification performance.

In the future, the proposed feature engineering framework will be extended by developing additional operational, route-level, and schedule-related features that capture more complex interactions between flight activities.

Declaration on the Use of Generative AI

The authors confirm that generative AI tools were used only for language and grammar enhancement. All research concepts, methodology, experiments, analysis, and conclusions were independently developed by the authors, who take full responsibility for the accuracy and integrity of the manuscript.

References

[1] L. Carvalho, A. Sternberg, L. M. Gonçalves, A. B. Cruz, J. A. Soares, D. Brandão, D. Carvalho, and E. Ogasawara, "On the relevance of data science for flight delay research: a systematic review," *Transport Reviews*, vol. 41, no. 4, pp. 499–528, 2021.

[2] 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.

[3] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140–150, 2019.

[4] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–6.

[5] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pp. 6679–6687, 2021.

[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[9] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794.

[10] N. Cahyana, S. Khomsah, and A. S. Aribowo, "Improving imbalanced dataset classification using oversampling and gradient boosting," in *2019 5th International Conference on Science in Information Technology (ICSITech)*, 2019, pp. 217–222.

[11] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[12] Y. Tu, M. O. Ball, and W. S. Jank, "Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern," *Journal of the American Statistical Association*, vol. 103, no. 481, pp. 112–125, 2008.

[13] J.-T. Wong and S.-C. Tsai, "A survival model for flight delay propagation," *Journal of Air Transport Management*, vol. 23, pp. 5–11, 2012.

[14] V. Natarajan, S. Meenakshisundaram, G. Balasubramanian, and S. Sinha, "A novel approach: Airline delay prediction using machine learning," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 1081–1086.

[15] H. Khaksar and A. Sheikholeslami, "Airline delay prediction by machine learning algorithms," *Scientia Iranica*, vol. 26, no. 5, pp. 2689–2702, 2019.

[16] 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, no. 1, p. 3385463, 2024.

[17] A. Ayaydın and M. A. Akçayol, "Deep learning based forecasting of delay on flights," *Bilişim Teknolojileri Dergisi*, vol. 15, no. 3, pp. 239–249, 2022.

[18] S. B. Jadhav and D. V. Kodavade, "Enhancing flight delay prediction through feature engineering in machine learning classifiers: A real time data streams case study," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 2s, pp. 212–218, 2023.

[19] M. D. Afrane, Y. Xu, L. Li, and K. Wang, "Flight delay prediction using random forest with enhanced feature engineering," in *SoutheastCon 2025*, 2025, pp. 1055–1056.

[20] M. D. Afrane, Y. Xu, and L. Li, "Enhancing flight delay prediction with network-aware ensemble learning," in *Database Engineered Applications*, ser. Lecture Notes in Computer Science, G. Bergami *et al.*, Eds., vol. 15928. Cham: Springer, 2025, pp. 109–121.

[21] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–6.

[22] J. Yi, H. Zhang, H. Liu, G. Zhong, and G. Li, "Flight delay classification prediction based on stacking algorithm," *Journal of Advanced Transportation*, vol. 2021, no. 1, p. 4292778, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/4292778

[23] S. A. A. AlBassam and S. D. N. AlShahrani, "Flight delay prediction: Evaluating machine learning algorithms for enhanced accuracy," *PLOS ONE*, vol. 20, no. 12, p. e0335141, 2025.

[24] P. Cerda and G. Varoquaux, " Encoding High-Cardinality String Categorical Variables ," *IEEE Transactions on Knowledge & Data Engineering*, vol. 34, no. 03, pp. 1164–1176, 2022.

[25] T. Kanan, A. Mughaid, R. Al-Shalabi, M. Al-Ayyoub, M. Elbes, and O. Sadaqa, "Business intelligence using deep learning techniques for social media contents," *Cluster Computing*, vol. 26, no. 2, pp. 1285–1296, 2023. [Online]. Available: https://doi.org/10.1007/s10586-022-03626-y

[26] M. E. Farfoura, I. Mashal, A. Alkhatib, and R. M. Batyha, "A lightweight machine learning methods for malware classification," *Cluster Computing*, vol. 28, no. 1, p. 1, 2025. [Online]. Available: https://doi.org/10.1007/s10586-024-04755-2

[27] K. Kiliç and J. M. Sallan, "Study of delay prediction in the us airport network," *Aerospace*, vol. 10, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2226-4310/10/4/342