Predicting Flight Delays at Tirana International Airport Using Machine Learning: A Comparative Study of Neural Networks, XGBoost, and Decision Trees

KRESHNIK VUKATANA[1], MARIUS BACI[2], OGERTA ELEZAJ[3]
[1] University of Tirana, Faculty of Economy,
ALBANIA

[2]Polytechnic University of Tirana, Faculty of Information Technology,
ALBANIA

[3]Bimingham City University,
UNITED KINGDOM

Abstract: - This study develops and compares machine learning models to predict flight delays at Tirana International Airport (LATI). Using Neural Networks (NN), XGBoost, and Decision Trees (DT) as a baseline model, the study classifies flight delays in real time from the moment of departure. The models predict arrival delays, utilizing operational, environmental, and historical flight data. The performance of the models was evaluated at various thresholds, with XGBoost showing the best balance of precision and recall, while Neural Networks excelled in capturing delayed flights at the cost of increased false positives. The results show the use of machine learning algorithms in the management of delays for airports and airlines. Future work could improve the models through incorporation of additional airport features that can be more specific, like the number of runways, air traffic control information, and weather components like thunderstorms, strong winds, and high-altitude weather along the path of the flight, many of which might improve delay prediction and operational decision-making and might be captured easily through standard weather data.

Key-Words: - Flight Delay Prediction, Machine Learning, Neural Networks, XGBoost, Decision Trees, Real-Time Classification, Airport Operations, Tirana International Airport, Predictive Modeling, Delay Management

Received: January 29, 2026. Revised: April 4, 2026. Accepted: June 6, 2026. Published: September 14, 2026.

1 Introduction

The aviation industry is expanding very quickly, as the number of passengers will almost triple from 4.6 billion to 12.4 billion, according to the ICAO Strategic Plan 2026-2050 [1], leading to a growing need for more airspace and larger capacities in the airports [2]. To manage this increase effectively, it's essential to expand airport capabilities. As Albania's aviation industry grows, the issue of flight delays- defined as departures that occur at least 15 minutes later than scheduled [3], [4]- is becoming more noticeable. These delays are hard to pin down due to various factors such as weather, airport operations, airline management, air traffic, and passenger issues [5], [6]. Airports face challenges from delays as they strain resources like runways and gates, putting added pressure on security and resource management. The delay of the flights increases the costs of airlines in sectors of maintenance, operation, and staffing mainly. Both business and leisure travelers experience disruptions in their travel plans. A deep understanding of these kinds of delays is very important for companies that operate in the insurance sector, as they can manage travel and flight delay insurance policies. Taking the European annual delay cost into consideration, estimated between 6.6 and 11.5 billion euros [7], it is very evident that a large economic issue is raised due to these delays that impact airport and airspace capacity planning [8].

This paper examines flight delay prediction at Tirana International Airport using three machine learning models, such as Decision Trees, used as a traditional baseline model, and deep learning algorithms such as Neural Network and XGBoost. The models are improved with training datasets that contain weather information and flight data in order to be suited for the operational needs of Tirana Airport and the flight routes that operate from Tirana. Data from Eurocontrol's R&D archives [9] are used in the study and contain location and time aspects in the feature engineering process.

The contribution of this study lies not in introducing new algorithms, but in how existing models are applied within a real operational context. By focusing on Tirana International Airport, the analysis integrates operational, environmental, and engineered variables into a predictive framework tailored to the conditions of the airport. In addition, the study deals with decision thresholds and their role in improving the detection of delays in real-time operational use.

2 Literature Review

Methods of flight delay prediction can be grouped into three categories: statistical models, simulation and empirical approaches, and machine learning-based models.

Statistical models are based on analyzing historical data in order to identify patterns and relationships between the variables in the dataset. Common approaches included are regression analysis, Bayesian networks, and Kalman filtering [6]. For example, [10] developed a Bayesian network model to predict delays in subsequent flights by examining factors that had a key influence. Furthermore, the study done by [11] which builds the work of [12] develops statistical models to estimate the economic impact of flight delays on European airlines. These models take into account different components in the costs that include fuel consumption, maintenance, and passenger compensation. In a similar way [8] provides a statistical analysis of departure delays and their financial impact for both European airlines and airports, indicating cost drivers and the way delays tend to propagate across operations. This study is used as one of the key references in this research. Complementing this direct-cost perspective, [30] highlights the hidden costs of unpunctuality, showing that delays can undermine customer loyalty and airline market share even when direct compensation costs are not incurred.

Simulation-based and empirical models tend to better understand how delays emerge and how they are propagated in the air transport system. These models replicate operations of aircraft, scheduling, and interactions between flights. For example, [13] proposes a model that is agent-based in order to capture reactionary delay propagation in the European air transport network, and in a similar way [14] developed an airport-centric queuing network model to analyze the impact of capacity reductions on delays. The approaches above are useful in order to understand the behavior of the system, especially in airports with complex and interconnected environments.

The machine learning-based delay prediction methods process a large volume of historical data to build predictive models to capture complex or nonlinear relationships. Supervised learning is commonly used where models are trained on labeled data to classify/predict delays. Popular algorithms for this include artificial neural networks, support vector machines, and decision trees. [15] applied regression-based models for delay forecasting, while [6] improved prediction accuracy using gradient boosting techniques. In a similar way [16] and [17] showed a strong performance using random forests and decision trees with weather-related features and [18] also showed that neural networks can achieve high accuracy when aircraft-specific information is included. More recently [19] Introduced a probabilistic tree-based model to improve the reliability of prediction. While these studies show strong predictive performance, most of them focus on large-scale or generalized airport systems, with less emphasis on smaller or capacity-constrained airports like the airport in Tirana.

While the study by [6] achieved strong results using gradient boosting classifiers, it focused on a generalized dataset with multiple airports included. In contrast, this study focuses specifically on Tirana International Airport and incorporates operational, environmental, and engineered variables that reflect its specific characteristics. The use of Neural Networks and XGBoost allows the model to capture non-linear relationships between these variables, while Decision Trees are used more as a baseline for comparison. In addition, this study places emphasis on decision thresholds and their impact on delay detection, aiming to support real-time operational decision-making. This gap motivates the methodology presented in the following section.

3 Methodology

3.1 Algorithms Description

3.1.1 Decision Trees

A Decision Tree is a supervised machine learning algorithm, which splits the data into subsets that are based on values of the features in order to classify whether a flight is delayed or not delayed. The metrics, such as Gini Impurity or Entropy guide the splitting process. Gini Impurity is calculated as:

$$Gini(t) = 1 - \sum_{i=1}^{k} {p_i}^2$$

where p_i is the proportion of samples in class i . A lower Gini value indicates higher purity. Similarly, Entropy is used for the measurement of uncertainty, and it is calculated as:

$$Entropy(t) = -\sum_{i=1}^{k} p_i \, log_2(p_i)$$

Decision Trees are very intuitive and relatively easy to interpret, but they are prone to overfitting, which can be mitigated by techniques like pruning or ensemble methods such as Random Forests [20].

In this study, the Decision Trees model is used as a baseline to provide a simple and interpretable comparison against more advanced models used in the study. They help capture basic relationships between operational and environmental variables, although their performance is limited when dealing with more complex and non-linear interactions present in flight delay data.

3.1.2 Neural Network

A Neural Network is a computational model used to capture complex and non-linear relationships between variables such as departure time, weather conditions, and previous delays [21]. The model is trained using backpropagation [22], , where the difference between predicted and actual values is minimized, and the Adam optimizer is used to improve convergence efficiency [23]. The ReLU activation function is applied in the hidden layers due to its effectiveness in handling non-linear patterns in the data [24]. The output of each neuron is computed as:

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

(1) Neural Networks Formula

where y is the output, x_i are the input features, w_i are the weights and b is the bias term, with f being the activation function.

The neural network in the study consists of three dense layers with 128, 64, and 32 neurons, respectively, combined with a dropout rate of 0.3 to lower the effect of overfitting. A learning rate of 0.001 is used for model training for 30 epochs. Considering the structure and the size of the dataset, the selected configuration was chosen in order to balance the complexity of the model and to generalize better. Early stopping was not applied as the model showed a stable convergence within the defined number of epochs.

3.1.3 XGBoost (Extreme Gradient Boosting)

XGBoost [25] is an ensemble learning method based on gradient boosting where decision trees are built sequentially in order to improve the accuracy of the prediction. For structured datasets, it is very effective and also captures complex interactions between the variables. The objective function used in XGBoost combines a loss function and a regularization term and can be written as:

$$\text{Objective Function} = \mathcal{L}(\theta) + \Omega(\theta)$$

(2) XGBoost Formula

XGBoost uses second-order optimization to improve the learning process and enhance model performance [26]. In this study, XGBoost is applied using a binary logistic objective function to classify flights as delayed or not. Tuning of these hyperparameters was done through iterative testing, and the final model was configured with a learning rate of 0.09, a max depth of 11, and 500 estimators in total. The settings were selected in order to achieve a balance between predictive performance and also for overfitting.

Table 1. The original data of flight information from Eurocontrol

ECTRL ID	ADEP	ADES	FILED-OFF BLOCK TIME	FILED ARRIVAL TIME	ACTUAL-OFF BLOCK TIME	ACTUAL ARRIVAL TIME	AC Type	AC Operator	Actual Distance Flown (nm)
19xxxxx	LATI	LIPZ	1/3/2016 4:25	1/3/2016 5:47	1/3/2016 4:14	1/3/2016 5:27	A319	AZA	421
19xxxxx	LATI	LIRF	1/3/2016 4:45	1/3/2016 5:57	1/3/2016 4:38	1/3/2016 5:45	A320	AZA	369
19xxxxx	LTBA	LATI	1/3/2016 5:35	1/3/2016 7:15	1/3/2016 5:27	1/3/2016 7:10	A319	THY	456

Source: created by the authors

3.2 Data Processing

Before applying the algorithms to the dataset, a pre-processing step is performed to transform the raw data into a format suitable for analysis (Figure 1). The flight information, extracted from Eurocontrol's R&D archive [9] covers each quarter of the year (March, June, September, and December) from 2016 to 2022 for all flights across Europe.

One aspect of dataset filtering consisted of filtering the flights that departed from andlanded at the airport of Tirana. As shown in Table 1, this refined dataset contains 18 columns and 54,728 rows. Other filtering consisted of excluding military aircraft, cargo flights, and operations because the study focuses only on commercial aviation. Key variables such as "OFF_BLOCK_DELAY" and "ARRIVAL_DELAY" were calculated as the difference between planned and actual departure and arrival times. A "YEAR_MONTH" variable was also created to capture time-related patterns.

Feature engineering was applied to enrich the dataset (Figure 2). The number of passengers per flight was estimated by linking aircraft type to seating capacity and applying a load factor of 80% [27].

Flight categorization into 'morning', 'afternoon', 'evening' and 'night' flights was part of the time-based features that were extracted. Also, a 'Day_of_the_week" variable was created in order to distinguish between weekdays and weekends. The frequency of flights per day was calculated as the number of flights departing from and arriving in Tirana.

Considering infrastructure changes at Tirana Airport throughout the studied period, a new variable called 'Gate_Expansion' was introduced that shows the increase in the number of gates from 7 to 17 in July 2022.

To account for delay propagation, a "PREV_ARRIVAL_DELAY" variable was constructed by linking consecutive flights with the same aircraft. If the arrival airport of one flight matched the departure airport of the next, the delay was transferred to the subsequent flight. A variable capturing congestion was also introduced in order to capture the operational pressure at the airport, and it was calculated based on the number of flights within a 30-minute time window. Weather data from Meteostat [28] was integrated by matching departure and arrival times with local weather conditions at Tirana Airport.

The dataset contains a higher proportion of non-delayed flights compared to delayed ones, indicating class imbalance. This is addressed in the evaluation stage by prioritizing recall and F1-score and by adjusting classification thresholds to improve the detection of delayed flights.

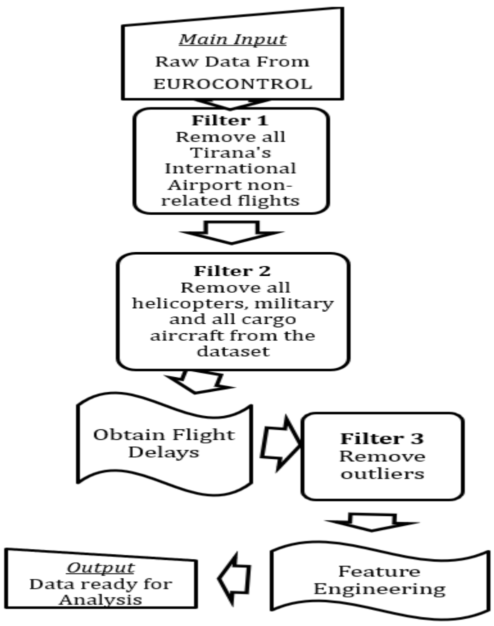

Fig. 1: Flowchart of the data filtering process
Source: created by the authors

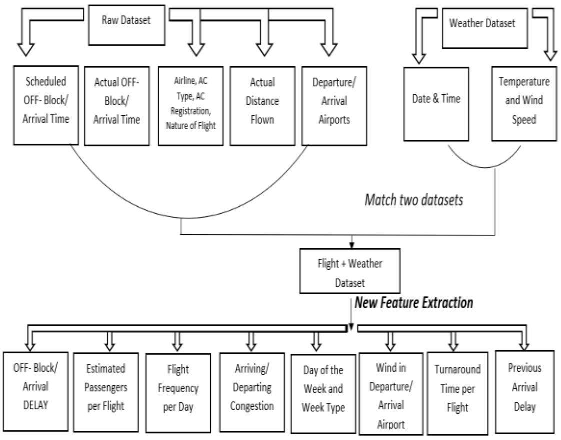

Fig. 2: Flowchart of feature engineering
Source: created by the authors

3.3 Model training & evaluation

To ensure a robust predictive performance and minimize generalization error, systematic model tuning and evaluation were performed prior to final model deployment, as shown in Figure 3.

To ensure reliable model performance, the dataset was split into training and testing sets using an 80%–20% ratio. A time-based split was applied in order to preserve the temporal structure of the data and better reflect real-world conditions where models are used to predict future flights based on observations from the past.

In order to improve the model performance and reduce overfitting, key parameters were tuned for each model. For the Neural Network model that is shown in Table 3, the architecture included three dense layers (128, 64, and 32 neurons) with a ReLU activation function and a dropout rate of 0.3 or 30%. Training of the model included the Adam optimizer with a learning rate of 0.001 for 30 epochs.

This configuration provided a very good balance between the complexity of the model and its generalization performance, and it was chosen after testing different setups.

No early stopping was applied, as the model showed stable convergence in the defined number of epochs used. This helps reduce overfitting and makes the model perform better on data that is unseen.

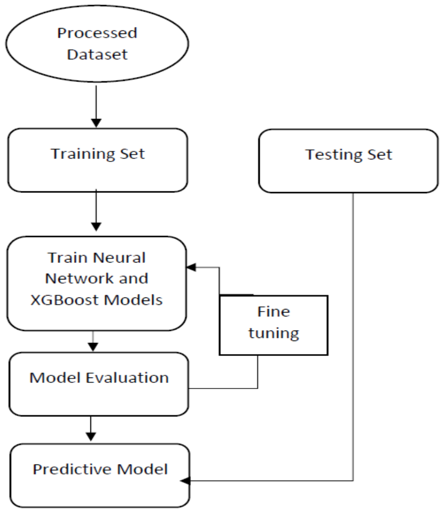

Fig. 3: Machine learning model workflow
Source: created by the authors

For XGBoost as shown in Table 2, hyperparameters such as learning rate, max depth, and number of estimators were tuned through iterative testing. The configuration in the final model consisted of a learning rate of 0.09, max depth of 11, and 500 estimators. These parameters were selected to capture complex feature interactions and to avoid the problem of overfitting.

Categorical variables such as ‘WEEK_TYPE’, ‘TIME_BLOCK’, ‘AC_TYPE’, ‘STATFOR_MARKET_SEGMENT’, ‘AC_OPERATOR’, ‘ADES’, and ‘ADEP’ were transformed using one-hot encoding to make them suitable for model training. The same pre-processing steps and features were applied to all models to ensure a fair comparison.

Missing values were handled differently across models. In Neural Networks, the missing values were treated implicitly during the training process, while XGBoost handled the missing values natively as ‘NaN’ values using its built-in mechanism.

Confusion matrices, precision, recall, F1 score, and ROC curves were used in order to evaluate the performance of the model. An emphasis was particularly put on recall and F1 score due to the imbalance of the class in the dataset and the importance of the operations in detecting delayed flights.

To further improve delay detection, different classification thresholds ranging from 0.0 to 1.0 were tested systematically. The selection of the optimal threshold was based on maximizing the F1-score while putting greater importance on recall.

The operational objective of identifying as many delayed flights as possible was achieved by the mentioned approach, even at the cost of higher false positives.

In addition to Neural Network and XGBoost, the Decision Tree model that is used as a baseline in the study was also tuned in order to provide a meaningful baseline comparison. The model was configured using the Gini criterion, with a maximum depth of 10 and class weights set to "balanced" to account for class imbalance. Additional parameters such as the number of estimators, random state, and minimum samples per leaf were selected to improve model stability and avoid overfitting. The full set of parameters used for the Decision Tree model is presented in Table 4.

Table 2. XGBoost Model Parameters

Hyper-parameter	Value	Hyper-parameter	Value
Learning Rate	*0.09*	**N_estimators**	*500*
Max Depth	*11*	**Min Samples Leaf**	*10*
Min Samples Split	*9*	**Min Samples Split**	*14*

Source: created by the authors

Table 3. Neural Network Model Parameters

Hyper-parameter	Value	Hyper-parameter	Value
Dropout Rate	*0.3*	**Learning rate**	*0.001*
Neurons (Layer 1)	*128*	**Neurons (Layer 2)**	*64*
Neurons (Layer 3)	*32*	**Epochs**	*30*

Source: created by the authors

Table 4. Decision Tree Model Parameters

Hyper-parameter	Value	Hyper-parameter	Value
Criterion	*Gini*	**Number of splits**	*500*
Max Depth	*10*	**Random state**	*42*
Class weight	*balanced*	**Min samples leaf**	*50*

Source: created by the authors

4 Results

The study developed predictive models to identify flight delays at Tirana International Airport (LATI) from the moment a flight takes off, using machine learning techniques, specifically Decision Trees, Neural Network (NN), and XGBoost.

These models focused on predicting/classifying arrival delays from the moment of a flight's departure.

The target variable 'ARRIVAL_DELAY_CLASS' classified flights as delayed if their arrival time was 15 minutes later than their scheduled time, which is a widely accepted benchmark for delay classification in aviation papers [4].

4.1 Model Comparison at Threshold = 0.5

All three models at a threshold of 0.5, including Neural Network, XGBoost, and Decision Tree, demonstrate distinct performances in their precision, recall, and overall accuracy in predicting/classifying flight delays at Tirana International Airport, as shown in Table 5.

The Decision Tree model is used as a baseline algorithm for comparison with more advanced Machine Learning algorithms like Neural Network and XGBoost, mainly due to its simplicity and easier interpretability. But despite its simplicity, the Decision Tree model can also capture non-linear relationships between its features and the target variable. This factor allows this algorithm to be easy to interpret and transparent in the model's prediction. However, Decision Trees are prone to the problem of overfitting, and generalization might be a struggle when there is high noise in the data or when the dataset is imbalanced, which might often be the case when dealing with flight delay data. As can be seen from Table 5, Decision Tree achieves an accuracy of 86.71% , precision for class 0 of 87%, precision for class 1 of 62%, recall for class 0 of 99%, and recall for class 1 of 7%, which indicates a poor performance.

The Neural Network model, on the contrary, achieves an accuracy of 88.21%, with a precision of 97% for class 0 (non-delayed flights) and a precision of 54% for class 1 (delayed flights), overall better than the Decision Tree algorithm. The high precision for non-delayed flights shows that the model is very much accurate when it predicts a flight as a non-delayed one. On the other hand, the low precision for delayed flights indicates that the model labels incorrectly a significant number of non-delayed flights as delayed, which leads to a high number of false positives.

Despite this, the model performs well in terms of recall for class 1, as it can correctly identify 86% of the delayed flights. On the other hand, the recall for class 0 is 89%, which shows that even though the model is fairly good, it still misses identifying some non-delayed flights. The F1 score of 66% for class 1 reflects the trade-off between precision and recall, as the model prioritizes identifying delayed flights, but this comes at the cost of more false positives.

At a treshold of 0.5, XGBoost model achieves a higher accuracy of 93.13% and a precision for class 0 of 95% and 79% for class 1. The high precision for both classes, especially for class 1, indicates that it makes more accurate predictions when classifying flights as delayed. It also achieves a higher recall of 97% compared to the algorithm of Neural Network and also of Decision Tree for class 0, which means it correctly identifies most of the flights that are non-delayed. The recall metric for class 1 is 68%, which is slightly lower than Neural Network, but still has a strong performance. This fact makes the XGBoost algorithm more balanced overall in identifying both the delayed and non-delayed flights with fewer false positives compared to the Neural Network algorithm.

Table 5. Performance Comparison of NN and XGBoost Models with Threshold=0.5

Metric	**Decision Trees**	**Neural Network (NN)**	**XGBoost**
Accuracy	86.71%	88.21%	93.13%
Precision (Class 0)	87%	97%	95%
Precision (Class 1)	62%	54%	79%
Recall (Class 0)	99%	89%	97%
Recall (Class 1)	7%	86%	68%
F1-Score (Class 0)	93%	93%	96%
F1-Score (Class 1)	12%	66%	73%

Source: created by the authors

In the confusion matrices below, it is seen that for the Neural Network in Figure 4 there are 8833 true negatives (non-delayed flights correctly predicted as non-delayed), 1347 true positives (delayed flights correctly predicted as delayed), but it misclassifies 1132 non-delayed flights as delayed (false positives), and it misses 228 delayed flights (false negatives). For the XGBoost model shown also in Figure 5, there are 9676 true negatives (non-delayed flights correctly predicted as non-delayed), 1071 true positives (delayed flights correctly predicted as delayed), with 289 false positives and 504 false negatives.

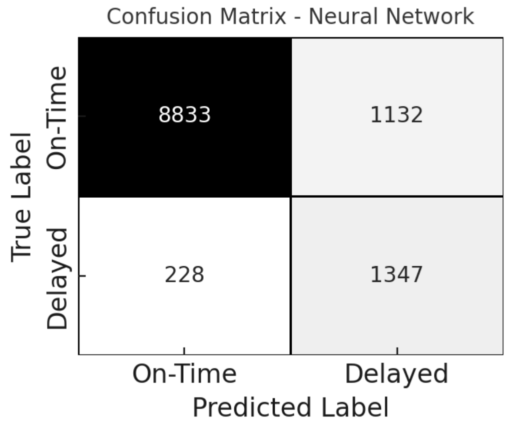

Fig. 4: Confusion Matrix for Neural Network Model with Treshold= 0.5
Source: created by the authors

Overall, at the default threshold of 0.5, XGBoost provides the most balanced performance across all classes, while the Neural Network model shows a stronger ability in the detection of delayed flights but at the cost of higher false positives. The Decision Tree model, which is simple and easy to interpret, performs poorly in identifying delayed flights, as reflected in its very low recall for class 1.

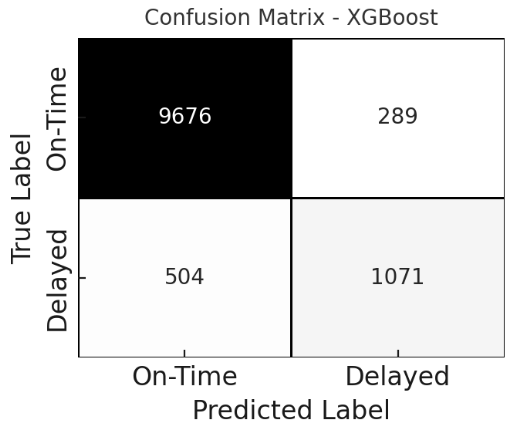

Fig. 5: Confusion Matrix for XGBoost Model with Treshold= 0.5
Source: created by the authors

4.2 Model Comparison at Threshold = 0.25

At the threshold of 0.25, in comparison with the 0.5 threshold, all three Decision Tree, Neural Network, and XGBoost models show a significant improvement in recall, especially for class 1 (for the delayed class), as shown in Table 6.

The threshold decrease makes the models more aggressive in predicting delays as they capture more delayed flights, but there is also an increase in the number of false positives. The adjustment is intentional in order to prioritize the detection of as many delayed flights as possible, as it is crucial for effective airport and airline operations.

The Decision Tree model did not perform very well. It showed an improvement of 29% recall for class 1, but at the expense of lower precision for both classes, especially for delayed flights. The precision for class 0 was 89%, but the precision for class 1 remained low at 32%, and the model still missed a substantial number of delayed flights.

For the Neural Network model, recall for class 1 increases to 93%, showing more delayed flights that are detected, but precision for class 0 (non-delayed flights) drops to 99%. False positives increase even with the precision for non-delayed flights remaining high, which shows that more on-time flights are classified incorrectly as delayed. The shift in the balance leans more towards recall, which aligns with the goal of early identification by airlines, especially in order to allow them to make proactive decisions.

In a similar way, XGBoost also sees an increase of 83% in recall for class 1, but at the cost of a little decrease of 97% in precision for class 0. XGBoost reaches a solid performance across classes as it achieves a better balance between precision and recall compared to the Neural Network and Decision Tree. However, lowering the threshold to 0.25 still leads to some false positives, though fewer than in the Neural Network or Decision Tree model.

Table 6. Performance Comparison of NN and XGBoost Models with Threshold=0.25

Metric	**Decision Trees**	**Neural Network (NN)**	**XGBoost**
Accuracy	81.58%	81.68%	91.78%
Precision (Class 0)	89%	99%	97%
Precision (Class 1)	32%	51%	66%
Recall (Class 0)	90%	80%	93%
Recall (Class 1)	29%	93%	83%
F1-Score (Class 0)	90%	88%	95%
F1-Score (Class 1)	30%	58%	73%

Source: created by the authors

From an operational perspective, putting recall as a priority is more important than maximizing the precision of the model because failing to detect a delayed flight (false negative) might lead to higher operational and financial costs to the airlines compared to false alarms. Therefore, a lower threshold provides a more practical solution for real-time delay management in environments with constrained capacities, such as Tirana International Airport.

4.3 ROC Curve and Feature Importance

The ROC curve that is shown in Figure 6 illustrates the performance of the XGBoost model. The curve plots the True Positive Rate or Recall on the y-axis against the False Positive Rate (1 - Specificity) on the x-axis. A larger area under the curve (AUC) shows a better performance for the model.

Although ROC/AUC is presented only for XGBoost, as it demonstrated the highest discrimination ability among the two other models, it is worth mentioning that similar performance patterns were also observed for Neural Networks and Decision Trees.

In this case, the AUC is 0.96, which demonstrates the ability of the model to distinguish between delayed and non-delayed flights. The steep ascent of the curve in Figure 6 shows that the model is good at identifying delays with a low rate of false positives in order to align with the goal of the study to put a priority on recall.

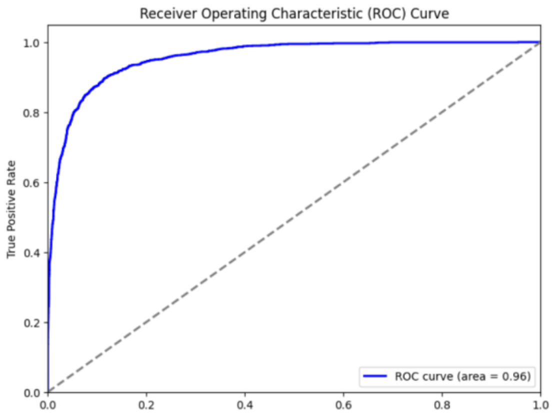

Fig. 6: ROC Curve
Source: created by the authors

The feature importance plot from the XGBoost model in Figure 7 shows that the most influential factors for predicting flight delays are linked to operational details and flight-specific variables. The distance flown by the aircraft is the most significant factor, followed by the delay experienced before the flight departs and the previous delay history of the flight.

These factors suggest that past delays and flight duration play a key role in predicting and classifying arrival delays in real time.

Other important features include flight frequency and weather conditions at Tirana, both of which directly influence delay predictions. According to [6], weather at departure and arrival airports is a major cause of delays, including factors such as low visibility, thunderstorms, and strong winds, while more complex effects, such as runway conditions are often difficult to capture in standard datasets [29]. Flight frequency and overall airport traffic also play a key role, aligning with findings by [8], particularly in capacity-constrained airports like Tirana. In addition, the congestion during scheduled arrival times and the estimated number of passengers further contribute to delays as higher traffic levels create operational pressure on airports.

These results confirm that, especially in a single airport study where the local conditions have an impact on performance, both operational and environmental variables are key drivers of delays. Also, feature importance provides useful insights, such as the non-linear relationships where interactions between variables can be complex and less straightforward to interpret. While there are no formal statistical significance tests conducted in the study, the consistency of results across models and thresholds suggests that the findings are robust.

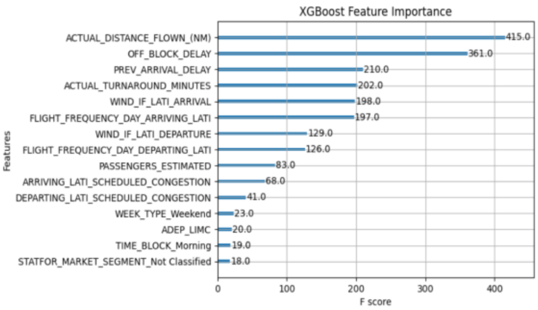

Fig. 7: Feature Importance Plot
Source: created by the authors

5 Conclusions

This study developed and compared machine learning models for predicting flight delays at Tirana International Airport, using Decision Trees as a baseline and Neural Networks and XGBoost as more advanced approaches. The models were designed to classify delays at the moment of departure by combining operational, environmental, and engineered variables derived from Eurocontrol data.

The results show that XGBoost provides the most balanced performance in terms of precision and recall, while the Neural Network is more effective in detecting delayed flights, although with a higher number of false positives. Lowering the classification threshold improves delay detection for both models, highlighting the importance of threshold selection in practical applications.

The main contribution of this study does not lie in introducing new algorithms, but in applying them in a real operational context. By focusing on a single airport and integrating feature engineering with threshold-based evaluation, the study shows how model configuration can support real-time decision-making. In this context, detecting delays early can be more valuable than minimizing false alarms, especially in environments with constrained capacities.

From an operational perspective, the findings suggest that prioritizing recall can improve delay management by increasing the detection of potential disruptions. This approach is relevant for airports with a limited capacity where even small delays can quickly propagate across the airport's operations.

Despite these findings, some limitations should be acknowledged. The study focuses on a single airport and relies on available operational and weather data, which may not capture all external factors affecting delays. Moreover, no formal statistical significance tests were conducted, although performance patterns remained consistent across models and thresholds.

Future work could extend this approach in order to include more detailed weather data, including en-route conditions, and apply AI techniques such as SHAP to understand the behavior of the model better. In addition to the above, expansion of the analysis to multiple airports or experimentation with hybrid and ensemble models could improve models in terms of generalization and predictive/classifying performance of the delays.

References:

[1] ICAO, "ICAO Strategic Plan 2026-2050," 2024. [Online]. Available: https://www.icao.int/about-icao/Council/Pages/strategic-plan-2026-2050.aspx[Accessed: 8 April 2025].

[2] European commission of mobility and transport, "Airport capacity and quality," 2017. [Online]. Available: https://ec.europa.eu/[Accessed: 12 May 2025].

[3] Eurocontrol, "Delays to air transport in Europe - Annual 2013," 2013. [Online]. Available: https://www.eurocontrol.int/. [Accessed: 22 April 2025].

[4] M. Jetzki, "The propagation of air transport delays in Europe," 23 12 2009. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2020-06/propagation-delays-2009.pdf. [Accessed: 26 February 2025].

[5] X. Geng, "“Analysis and countermeasures to flight delay based on statistical data”," in 5th Int. Conf. on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 2013. DOI: 10.1109/IHMSC.2013.275
[6] M. Lu, P. Wei, M. He and Y. Teng, "Flight Delay Prediction Using Gradient Boosting Machine Learning Classifiers," Journal of Quantum Computing, Tech Science Press, vol. 3, no. 1, pp. 1–12, 2021. DOI:10.32604/jqc.2021.016315.
[7] Institut du Transport Aerien, "Cost of air transport delay in Europe," 2000.
[8] J. Cardenas, I. d. Alvear, T. Haegens, T. Hendrix and F. Kerstens, "Aircraft Departure Delay and its Associated Costs for Airlines and Airports in Europe," Research Gate, 2017. https://www.researchgate.net/publication/330989073_Aircraft_Departure_Delay_and_its_Associated_Costs_for_Airlines_and_Airports_in_Europe. [Accessed: 10 January 2025].
[9] EUROCONTROL OneSky Online Extranet, 2024. [Online]. Available: https://ext.eurocontrol.int/elsh/private/getOnlineServicesList.do. [Accessed: 17 February 2025].
[10] Q. Li, L. Wang, R. Fei, B. Wang and X. Hei, "An analysis method for flight delays based on Bayesian network," in The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China., 2015. DOI: 10.1109/CCDC.2015.7162353
[11] University of Westminister, "European airline delay cost reference values," EUROCONTROL, 2015.
[12] A. Cook, "European Air Traffic Management - Principles, Practice and Research," Burlington, USA, Ashgate publishing company, 2007. ISBN: 978-0-754-67295-1.
[13] B. Campanelli and P. Fleurquin, "Modeling Reactionary Delays in the European Air Transport Network," in SIDs 2014 - Proceedings of the SESAR Innovation Days, Madrid, Spain, 2014.
[14] B. Baspinar, N. Ure, E. Koyuncu and G. Inalhan, "Analysis of Delay Characteristics of European Air Traffic through a Data-Driven Airport-Centric Queuing Network Model," in IFAC-PapersOnLine, 2016. DOI: 10.1016/j.ifacol.2016.07.060
[15] E. Šimić and M. Begović, "Airport delay prediction using machine learning regression models as a tool for decision making process," in 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia,2022. DOI: 10.23919/MIPRO55190.2022.9803809
[16] N. Chakrabarty, "Flight arrival delay prediction using gradient boosting classifier," in Emerging Technologies in Data Mining and Information Security, Springer, Singapore, 2019. DOI: 10.1007/978-981-13-1498-8_57
[17] S. Choi, Y. J. Kim, S. Briceno and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in IEEE/AIAA 35th Digital Avionics Systems Conference, Sacramento,CA, 2016. DOI : 10.1109/DASC.2016.7777956
[18] E. Demir and V. B. Demir, "Predicting flight delays with artificial neural networks: Case study of an airport," in 25th Signal Processing and Communications Applications Conference, Antalya, 2017. DOI: 10.1109/SIU.2017.7960463
[19] R. Dalmau, "Probabilistic and explainable tree based models for rotational reactionary flight delay prediction," CEAS Aeronautical Journal, pp. 1157-1173, 2024. DOI: 10.1007/s13272-024-00750-w
[20] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. DOI:10.1023/A:1010933404324.
[21] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," Nature, vol. 521, Issue 7553, pp. 436-444, 2015. DOI:10.1038/nature14539.
[22] D. Rumelhart, G. Hinton and R. J. Williams, "Learning representations by back-propagating errors.," Nature, pp. Volume 323, Issue 6088, Pages 533-536, 1986. DOI:10.1038/323533a0
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 2015.
[24] X. Glorot, A. Bordes and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, , 2011. Proceedings of AISTATS, PMLR, vol. 15, pp. 315-323
[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, pp. 785-794, 2016. DOI:10.1145/2939672.2939785.
[26] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," The Annals of Statistics, Volume

29, Issue 5, Pages 1189-1232, 2001. DOI:10.1214/aos/1013203451.
[27] Eurocontrol, "Standard Inputs for Eurocontrol Cost-Benefit Analyses," January 2018. [Online]. Available: https://www.eurocontrol.int/sites/default/files/publication/files/standard-input-for-eurocontrol-cost-benefit-analyses-2018-edition-8-version-2.6.pdf. [Accessed: 24 March 2025].
[28] Meteostat, 2025. [Online]. Available: https://meteostat.net/en/station/13615?t=2025-04-04/2025-04-04. [Accessed: 1 March 2025].
[29] M. Güvercin, N. Ferhatosmanoglu and B. Gedik, "Forecasting flight delays using clustered models based on airport networks," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 5, pp. 3179–3189,2020. DOI:10.1109/TITS.2020.2990960.
[30] A. Cook, G. Tanner and A. Lawes, "The hidden cost of airline unpunctuality," Journal of Transport Economics and Policy, vol. 46, no. 2, pp. 157-173, 2012. DOI: 10.3828/jtep.2012.46.2.157

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US