CSE 6242 Final Report - Team 8 (Super Fly)

Nicola Ciocchini, Yusheng Xia, Makram Harmoush, Suresh Sivakumaran, Rutvij Rana, Xiaojie Yu

1 INTRODUCTION

In an era where travel efficiency is paramount, passengers are often torn between choosing cost-effective flights and opting for routes with a higher on-time performance. Existing platforms, such as Google Flights, prioritize fare cost without accounting for reliability of flight routes. This project aims to revolutionize how passengers select flights by introducing a recommendation engine that integrates both price and reliability metrics. Utilizing a blend of historical and real-time data, our engine presents a dynamic solution that caters to the modern traveler's needs.

2 PROBLEM DEFINITION

Navigating the complexities of flight selection, passengers often grapple with the decision on whether to prioritize lower fares or more reliable routes. The lack of comprehensive platforms that consider both cost and flight reliability leaves a significant gap in the decision-making process. Our engine addresses this gap by providing a nuanced recommendation system, thereby enhancing the travel planning experience.

3 LITERATURE SURVEY

Our review encompasses diverse methodologies for predicting flight delays and prices, setting the foundation for our innovative approach. Key studies have demonstrated the efficacy of deep learning, probabilistic forecasting, and network analysis in enhancing prediction accuracy.

[1] Yazdi et al. (2020) employed a stack Denoising Autoencoder SDAE to extract features from flight-related data, thereby improving the accuracy of a Levenberg-Marquart algorithm. They factored in weather, airlines and airports and generated models with improved prediction accuracy. Additionally, deep learning can be used for more complex flight patterns.

[2] Carvalho et al. (2021) provided an overview of flight delay prediction research. They analyzed aspects of prediction models, data sources, features, and algorithms. They also reviewed existing approaches' challenges to guide the model selection (regression, classification, ensemble methods). Based on their results, modeling may be improved by blending different models.

[3] Shah et al. (2020) introduced a two-stage model for optimizing flight delay prediction: first, a binary classification to predict delay occurrence; and then, evaluating classifiers (e.g., SVM, RF) for accuracy. This highlights the importance of employing ML techniques and integrating real-time data into the model for accuracy.

[4] Kim et al. (2016) showed that recurrent neural networks (RNN) could predict delays in 10 airports with +85% accuracy by considering sequential flight influences. Based on their findings, it may be necessary to incorporate scalability to larger datasets to address real-time disruption's limitations.

[5] Li et al. (2022) utilized complex network theory to extract spatial features of the aviation network. By combining random forest with RNN, they achieved an accuracy of +90% for delays in airports. This showed the importance of spatial features for predicting delays and encouraged us to consider the impacts of diverse airports in the aviation network.

[6] Cai et al. (2021) used graph convolutional neural network (GCN) to develop a time-evolving airport network for flight delay prediction. Spatial interactions were also considered. This further highlighted the importance of spatial elements. Therefore, spatial features of North America can be used to improve our project.

[7] Ding (2017) developed a multiple linear regression model to predict flight delays and achieved better accuracy than Naive-Bayes and C4.5 approach with real-world data. This supported the feasibility of a simple and practical prediction model for flights. This model, along with exploration of other models, could help enhance computational efficiency.

[8] Gui et al. (2019) investigated parameters and models affecting flight delays, finding that a random forest-based model can achieve $\tilde{9}0\%$ accuracy and mitigate overfitting. The team can account for overfitting while integrating with other models.

[9] Qu et al. (2020) combined meteorological and flight data to develop a deep convolutional neural network for flight prediction, achieving +90% accuracy. The team's integration of additional data sources like meteorological data may improve prediction accuracy.

[10] Wang et al. (2022) crafted a causal flight delay prediction model accounting for direct/indirect factors and an attention mechanism. This proves the importance of peripheral mechanisms for our model.

[11] Belcastro et al. (2016) used parallel algorithms (i.e. MapReduce) to assess how weather affects flight delays, processing airline and weather data. While challenging to implement, we can explore alternative approaches to incorporate this parameter's complexity.

[12] Jin & Sendhoff (2003) tackled trade-offs via multi-objective optimization, balancing performance and robustness. This aligned with our project of balancing reliability and price. Using variance-based methods for measure selection could build on their work.

[13] Verma & Sharma (2022) analyzed flight pricing dynamics, identifying economical and luxurious airline categories and factors like departure time, day, and holidays affecting airfare. This should be considered to improve price accuracy.

[14] Sahadevan et al. (2021) predicted scheduled flights' landing times by addressing time fluctuations with two models: an experience-based one and a nonlinear regression model. This could help improve our project's understanding of flight reliability if random flight variations are identified.

[15] Yu et al. (2019) utilized a multifactor approach to predict flight delay patterns and generated a novel prediction model based on deep belief network (DBN). They refined the model with supervised support vector regression (SVR), showcasing the combination of two models in the same domain, which could guide our team, as well.

[16] Esmaeilzadeh et al. (2020) applied an SVM model to study the non-linear link between flight delays. They found that factors such as pushback delay, taxi-out delay, ground delay program, etc. are key contributors to delays. This showed the significance of considering factors like these in our data source.

[17] Mamdouh et al. (2023) presented FDPP-ML, leveraging past flight paths for improved delay prediction. By integrating machine learning, it outperformed other methods across 10 different models. This also opened the window for our team to explore integration possibilities.

[18] Tziridis et al. (2017) focused on predicting airfare prices using ML models (i.e. regression tree, etc.) with an 88% accuracy. It's useful to understand the influence of flight characteristics on pricing. For this approach to be used, it needs to be more generalized.

4 PROPOSED METHOD

4.1 Overview

Our flight recommendation engine is designed to redefine the flight selection process, offering a user-centric platform that melds price comparison with reliability assessments. The integration of real-time data combined with dynamic modeling of historical data ensures up-to-date recommendations, while our focus on an intuitive interface promises an accessible user experience.

4.2 Innovation

Our approach is characterized by four primary innovations:

1. **Dual-Focused Flight Recommendations**: By considering both price and reliability, our engine provides a more comprehensive assessment of flight options that has not been achieved yet by any other platform.
2. **Dynamic, Real-Time Modeling**: Unlike traditional systems that rely on static data, our engine continuously updates its predictions and recommendations based on real-time data and changing conditions.
3. **Predictive Insights into Flight Reliability**: Our model goes beyond reporting historical delays only; it anticipates both the likelihood and potential duration of delays for specific flights, offering a predictive edge that allows users to plan with greater confidence.
4. **Interactive, User-Friendly Interface**: Our platform makes complex data accessible and engaging, allowing users to explore flight options with ease and gain clear insights into the trade-offs between cost and reliability.

By addressing these key areas, our project introduces a new standard of travel planning, where transparency, reliability and user empowerment are at the forefront.

4.3 Implementation Highlights

The development of our flight recommendation engine is grounded on a blend of cutting-edge computational techniques and user-centric design principles, namely:

Real-Time Data Library: By leveraging API data connections to TripAdvisor's online data repository via RapidAPI, users of our platform are able to obtain estimates of flight delays for their particular flight that are based on the latest available flight patterns along with flight prices. This means detailing trends that stem from their airline, airport, destination, time of departure, and other leading indicators. Through having access to a library that constantly refreshes with live data, we ensure that our platform provides users with the most up-to-date information on their flight status estimates.

Probabilistic Modeling: Through our access to historical data across all flights from Kaggle, we are able to provide estimates on delay likelihood and forecast delay for any US based airline flying from a particular departure city to arrival city. Connecting flights have their delay probabilities chained using the product rule so that users are able to obtain a holistic estimate across all their flights, a view that does not currently exist today in real time. This algorithm allows users to understand the historical reliability of their airline at different airports. This means users are able to select a threshold delay criteria (eg. 2+ hours) and be provided with a single probability of delay value. Criteria for price can also be layered on to these predictions such that users can ensure the flight recommendation does not exceed a certain price while ensuring maximum reliability.

ML Predictors on Estimated Delays: While simply understanding the probability of delays is one potential method of understanding airline performance, our platform aims on expanding that view to include machine learning predictions on what the estimated delay time would be based on numerous indicators provided by the user. These factors include airline, origin city, destination city, time of departure, distance, and seasonality factors such as year, month, and day. A boosted decision tree regression model trained on historical flight data will be used to provide flight delay expectations for any one-way flight, with the model continuously updating as newer Trip Advisor data comes in. Flight delay estimates could also include confidence intervals so that delay estimates will also contain a range of values for users who need a bit more accuracy when planning their trip. With these outputs, users will be able to review potential flight routes with expected delay times and flight prices to best plan for their trip.

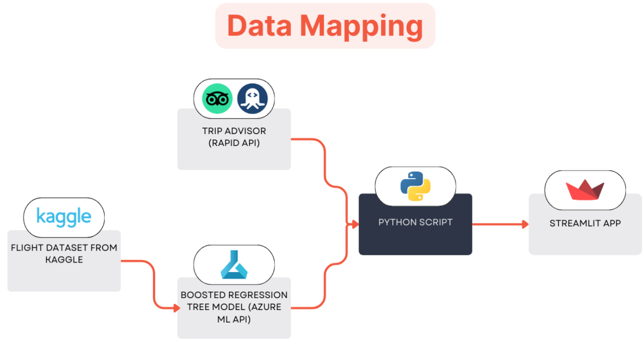

Figure 1: Visual of data pipeline.

User Friendly Interface: At the forefront of our platform is a user-centric interface that simplifies the exploration of flights and enables travellers to input their preferences and constraints easily. At minimum the platform will generate three outputs: 1) Probability of delay, 2) Estimated delay, and 3) Up-to-date Price. It should be easy to use by individuals and allow for the ability to input complex flight patterns that include multi-stop flights across multiple airports and airlines.

5 EXPERIMENTATION/EVALUATION

Multiple iterations were done before reaching final implementation results. Decisions on what the concluding model selection, UI integration, and product offering were informed based on results from initial tested questions across a variety of bucket categories as detailed in the following sections.

Real-Time Data Library: Question: Is it possible to connect live TripAdvisor Data to UI Platform in order to provide end-users with up-to-date flight information and provide models with historical training data?

The team has successfully implemented the API connection to TripAdvisor's flight data repository via RapidAPI and is able to receive live flight data and estimates with high accuracy. This enables the team to pull in real-time availability of flights across source and destination cities along with current prices, thereby successfully integrating one portion of the real-time recommendation engine.

Multiple tools were used to try experimenting the connection but the end result the team used was a python script that was deemed most efficient in maintaining the API connection with the UI platform for end-users.

Probabilistic Modeling: Question: Are we able to leverage historical flight information on delays to provide end-users who have knowledge on their airline, departure city, destination city, and date information a likelihood that their flight will suffer delays?

The team has successfully implemented a probabilistic model detailing the probability of flights being delayed past a certain threshold given an airline, origin city, and destination city. The product rule has been applied to multi-stop flights and the results provide a clear indication of the historical probability density of delays in the data subset.

Due to the complexity of incorporating edge-case scenarios and custom flight patterns, the team ultimately limited rules to only provide simplistic arrival delay estimates that would sum multi-step flight delays. A python script was embedded to allow for historical data loading and probability calculations. Source code detailing methodology were included in the package that helped detail the proof of concept.

However, due to overlap with the ML model which provided a more holistic point of view on total delay expectations while achieving higher accuracy on probability of delay (logistic regression model outputs leveraging Azure ML) the final implementation of the probability based modeling was excluded from the UI.

ML Predictors on Estimated Delays: Question: What is the most optimal ML model that provides accurate flight arrival delay results as well as integrates efficiently and effectively with the UI output.

The team experimented with 4 different ML models to understand the accuracy of predicted flight delays. In order to reduce noise and limit the scope of the analysis, the team focused primarily on returning predicted arrival delays for one way direct flights. The original attempts leveraged python scripts along with the sklearn models. Given the large size of the dataset and computational complexity of a random forest model, the team experimented with training on subsets or batch training to obtain model results and compare options. While results were decent, training time on local computer was a major obstacle to further implementation, as options such as Grid-Search hyperparameter tuning were too computationally expensive. Secondary attempts were done using Azure's ML platform, and showed initial promising results.

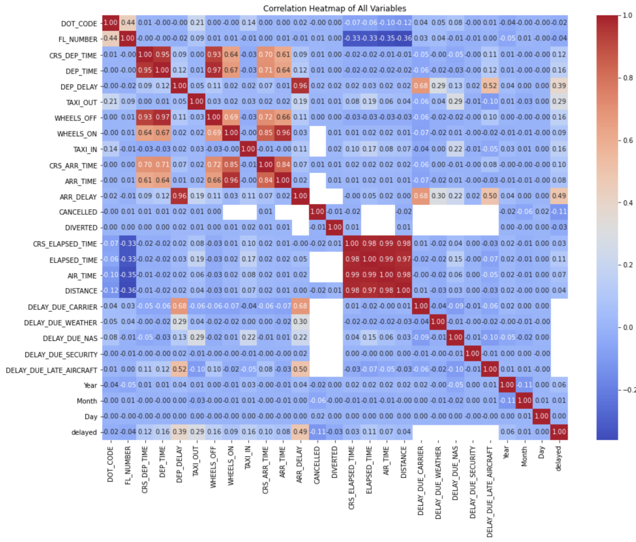

Figure 2: Feature selection analysis.

Further experimentation was thus done on feature selection. In short, the team spent time understanding which variables were highly correlated (Figure 2) or were infeasible to obtain when sourcing information from users in the UI stage (example, taxi time, etc). In the third and final phase, the team evaluated model performance based on statistical metric evaluations such as Mean Absolute Error (MAE) and Mean Squared Error (RMSE). Residual distribution was also illustrated (Figure 3) to best understand the distribution of errors and what that meant for end users. Based on the performance results, the team decided to choose Boosted forest regression model to implement the prediction UI.

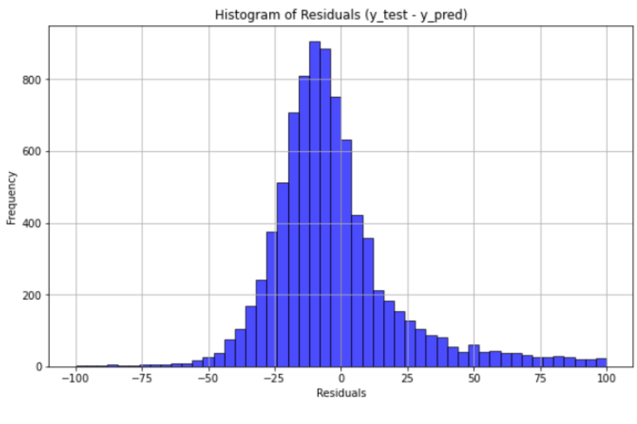

Figure 3: Residual Distribution analysis.

Models explored:

1. **Random forest ensemble (subsampled in python)**: Due to the size of our dataset , we use a method called ensemble learning where we break out train-

ing set into many smaller models (random sampling) and then average out results. This works for random forest regressors due to the complexity which can be overwhelming when we have a lot of data. Our random forest model indicates a Minimum Absolute Error (MAE) for estimated delay of 19.9 minutes which implies a moderate level of accuracy. Served as basis for replicating modeling in Azure.

2. **Linear regression (subsampled in python)**: Facing similar constraints with size, we implemented a subsampled multiple linear regression predictor through Python that produced a MAE of 20.8 minutes. Served as basis for replicating modeling in Azure.
3. **Linear and logistic regression (full dataset in Azure)**: To overcome size limitations and utilize an intuitive graphical user interface in building our ML model, we utilized the Azure Machine learning model builder in exploration 3. This method allowed for training of the full dataset and was computationally much more efficient. Initial runs showed overly optimistic results due to confounding variables but post feature selection, the model returned better test set results when compared with the subsampled python run. We conducted experiments using Linear Regression and Boosted Forest Regression with all columns, with the target column as arrival delay. Additionally, we employed Linear Regression and Boosted Forest Regression with the the relevent features or subset of columns as our independent variables: flight date, airline name, airline code, flight number, origin, destination, air time, distance traveled, and if a flight was canceled, while arrival delay remained the dependent variable. Further tests were performed using the same subset of columns for both independent and dependent variables, employing Multiclass Logistic Regression, Two-Class Logistic Regression with a 0.5 threshold, and a Two-Class Decision Forest with a 0.5 threshold. Most of the models presented positive results (refer to Figure 4), yet the selection of the model was determined by its relevance to our objectives.
4. **Boosted Decision Tree Regression (full dataset in Azure)**: The team ultimately landed on using a boosted decision tree regression, which showed the most promising accuracy results after train/test implementation. The ability to build trees sequentially that corrected mistakes without too high of a cost of overfitting in our data scenario led to superior results when compared to random forest regression, and the outputs of the model were great in that it provided an arrival delay time estimate that was useful for user outputs. In addition, running the training on Azure nulled the potential risk of long training times that typically come with Boosted Decision Trees vs Random Forests. To further refine our approach, we decided to ultimately choose the trained models with our selected features or subsets, instead of the training data with all the columns. This decision was driven by the recognition that the full dataset was not fully representative and was overfit for arrival delay. We also opted for a linear regression approach to provide an arrival delay compared to the logistic regression options that would provide a probabilistic delay distribution. Despite these considerations, the Boosted Decision Tree model showed a Mean Absolute Error (MAE) of 17.44, which was the lowest of our performance within the subset of data. The final model was built as a real-time inference pipeline on Azure and can be seen in Figure 5.

Performance Metrics	Independent variables are all columns; arrival delay as dependent		Independent variables are subset of columns; arrival delay as dependent	
	Linear Regression	Boosted Forest Regression	Linear Regression	Boosted Forest Regression
MAE (Mean Absolute Error)	0.025136	2.067141	18.016774	17.439848
RMSE (Root Mean Squared Error)	0.294626	4.763171	44.697694	44.387381
RSE (Relative Squared Error)	0.000033	0.008614	0.75858	0.748084
RAE (Relative Absolute Error)	0.001047	0.08611	0.750519	0.726486
R2 (Coefficient of Determination)	0.999967	0.991386	0.24142	0.251916

Independent variables are subset of columns; arrival delay as dependent	
Performance Metrics	Multiclass Logistic Regression
Overall accuracy	0.925469
Micro_Precision	0.925469
Macro_Precision	0.929998
Micro_Recall	0.925469
Macro_Recall	0.906356

Performance Metrics	Independent variables are subset of columns; arrival delay as dependent	
	Two Class Logistics Regression (0.5 Threshold)	Two Class Decision Forest (0.5 Threshold)
Accuracy	0.927	0.947
Precision	0.953	1
Recall	0.836	0.851
F1 Score	0.891	0.919
AUC	0.982	1

Figure 4: Performance of various models. Boosted forest regression model selected (red border).

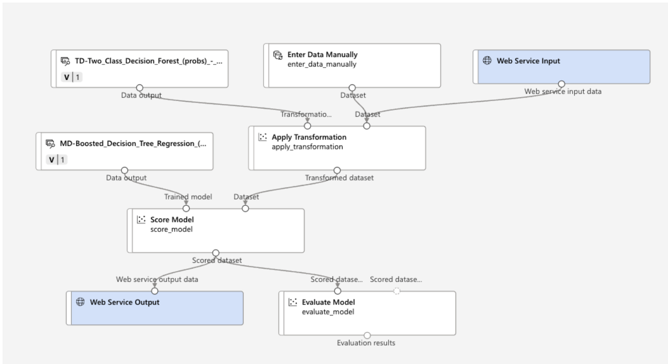

Figure 5: Real-time inference pipeline in Azure ML

User Friendly Interface: The user interface of the Flight Delay Tool is designed for ease of use and accessibility with the following structure deployed:

- **Input Menu**: Upon accessing the app, users will be greeted with an intuitive homepage that outlines the purpose and functionality of the Dual-Focused Flight Recommendation Engine. Users select the flight origins and destination and a list of the top flights available will be presented ranked by your choice of price, departure time, arrival time, and duration. Each option is displayed in a user-friendly format, allowing users to quickly scan through the list and identify relevant information.
- **Results page**: Upon selecting a specific destination, users are presented with a comprehensive tool designed to aid in making informed decisions regarding their travel plans. This tool includes an Expected Delay vs Price graph, which visually highlights any measurable differences within different carriers. Users can easily discern various flight options and their corresponding expected delays, enabling them to select flights that align best with their preferences and priorities simply by clicking on the corresponding data points. Additionally, a detailed list of flights, along with their respective information and delays, is provided below the graph. This information is derived from historical patterns and predictive insights, offering users a comprehensive view of available flight options. Negative delay values indicate flights that are expected to arrive earlier than scheduled, while positive delays indicate flights that are anticipated to arrive later than expected. The structured presentation of this information allows users to interpret delay data efficiently, empowering them to make well-informed decisions. Whether it involves selecting alternative flight routes or adjusting departure times, users can confidently plan their travel itineraries with clarity and precision.
- **User Interaction**: The interface empowers users to engage with the displayed delay data, providing them with the capability to filter, sort, or search for specific destinations according to their preferences. Below, you will find a depiction of the app interface, showcasing a graph along with options for a list of flights (Figure 6).

6 CONCLUSION

The main focus of the project stemmed on building a strong proof of concept that could then be expanded to

Figure 6: Interface of Superfly App

provide actual value to end users. There exists a lot of untapped value add when leveraging historical data on flights trained through machine learning and customized to fit individual user situations. The product can help many users understand the risks they undertake when deciding on the airline and flight they want to take. Expanding the UI and models to handle additional cities, multi-stop flights, and additional feature expansions are potential next steps that can greatly enhance the product's value-add to end users down the line. It's worth noting that for the current iteration, we've focused solely on one-way flights and economy class of service. However, this is just the beginning, and there's immense untapped potential to incorporate additional features in the future. Integrating options for multi-stop flights, premium classes, and other travel preferences could further enrich the user experience and provide even greater value. All members have equally contributed to the project. The GANTT chart shows initial work distribution and the final table shows finalized contributions.

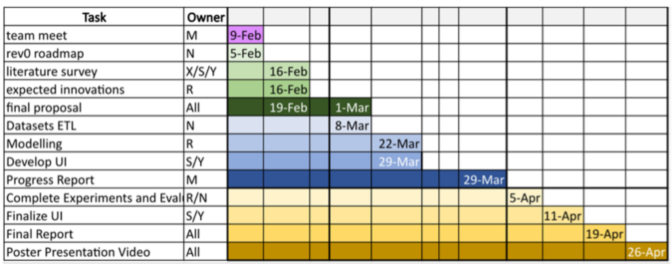

Figure 7: Gantt chart for project plan.

- Nicola (N) – data pipeline, modelling (20h)
- Rutvij (R) – modelling, feature selection (20h)
- Yusheng (Y) – modelling, reporting (20h)
- Siva (S) - user interface (20h)
- Xiaojie (X) – compiling data & report (20h)
- Makram (M) - user interface & report (20h)

7 BIBLIOGRAPHIES

1. Yazdi, M. F., Kamel, S. R., Chabok, S. J. M., & Kheirabadi, M. (2020). Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. *Journal of Big Data*, *7*, 1-28.

2. Carvalho, L., Sternberg, A., Maia Goncalves, L., Beatriz Cruz, A., Soares, J. A., Brandão, D., ... & Ogasawara, E. (2021). On the relevance of data science for flight delay research: a systematic review. *Transport Reviews*, *41*(4), 499-528.

3. Shah, D., Lodaria, A., Jain, D., & D'Mello, L. (2020). Airline Flight Delay Prediction Using Machine Learning Models. *International Journal of Recent Technology and Engineering* (IJRTE) ISSN: 2277-3878 (Online), Volume-9 Issue-2.

4. Kim, Y. J., Choi, S., Briceno, S., & Mavris, D. (2016, September). A deep learning approach to flight delay prediction. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* (pp. 1-6). IEEE.

5. Li, Q., & Jing, R. (2022). Flight delay prediction from spatial and temporal perspective. *Expert Systems with Applications*, *205*, 117662.

6. Cai, K., Li, Y., Fang, Y. P., & Zhu, Y. (2021). A deep learning approach for flight delay prediction through time-evolving graphs. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 11397-11407.

7. Ding, Y. (2017, August). Predicting flight delay based on multiple linear regression. In *IOP conference series: Earth and environmental science* (Vol. 81, No. 1, p. 012198). IOP Publishing.

8. Gui, G., Liu, F., Sun, J., Yang, J., Zhou, Z., & Zhao, D. (2019). Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*, *69*(1), 140-150.

9. Qu, J., Zhao, T., Ye, M., Li, J., & Liu, C. (2020). Flight delay prediction using deep convolutional neural network based on fusion of meteorological data. *Neural Processing Letters*, *52*, 1461-1484.

10. Wang, F., Bi, J., Xie, D., & Zhao, X. (2022). Flight delay forecasting and analysis of direct and indirect factors. *IET Intelligent Transport Systems*, *16*(7), 890-907.

11. Belcastro, L., Marozzo, F., Talia, D., & Trunfio, P. (2016). Using scalable data mining for predicting flight delays. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(1), 1-20.

12. Jin, Y., & Sendhoff, B. (2003, April). Trade-off between performance and robustness: An evolutionary multiobjective approach. In *International conference on evolutionary multi-criterion optimization* (pp. 237-251). Berlin, Heidelberg: Springer Berlin Heidelberg.

13. Verma, P., & Sharma, K. (2022). Flight Price Prediction. Jaypee University of Information Technology, Solan, H.P.

14. Sahadevan, D., Ponnusamy, P., Gopi, V. P., Guruswami, S., & Krishna, A. K. (2021). A machine learning-based approach to predict random variation in the landing time of scheduled flights. *International Journal of Sustainable Aviation*, *7*(4), 293-318.

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

16. Esmaeilzadeh, E., & Mokhtarimousavi, S. (2020). Machine learning approach for flight departure delay prediction and analysis. *Transportation Research Record*, *2674*(8), 145-159.

17. Mamdouh, M., Ezzat, M., & A. Hefny, H. (2023). A novel intelligent approach for flight delay prediction. *Journal of Big Data*, *10*(1), 179.

18. Tziridis, K., Kalampokas, T., Papakostas, G. A., & Diamantaras, K. I. (2017, August). Airfare prices prediction using machine learning techniques. In *2017 25th European Signal Processing Conference (EUSIPCO)* (pp. 1036-1039). IEEE.