# Predict Lost Flights Connections. An Interpretable Machine Learning Approach.

Hugo Miguel Silva Lopes
Instituto Superior Técnico, Universidade de Lisboa, Portugal
hugo.m.lopes@tecnico.ulisboa.pt

***Abstract*—In airlines, flight schedule optimization and passenger satisfaction are problems that profoundly impact the airline industry revenue every year. Missed connections are often a consequence of unexpected disruptions and the lack of preventive mechanisms that affect airlines' regular operations and image. This paper proposes a new approach for models to classify the success of passengers' connections through an airline hub, focusing on interpretability. This issue is key to airline profitability since decision-makers often want to have hard evidence before taking action. The models were trained on data from TAP Air Portugal's passenger activity from 2019 and the beginning of 2020, along with some data from airport movements. We analyzed the data and did some feature engineering, including encoding some features and generating new samples to re-balance the dataset. In total, we studied five models, two non-interpretable plus three interpretable models. The overall accuracy of the interpretable models was not as good as the results from the non-interpretable models. However, when looking for critical metrics for imbalanced data, as this is the case, and the performance on the minority class, i.e., missed connections, the interpretable models had a performance close to the one seen in the best non-interpretable model. These metrics included the Recall on the minority class and the macro-average Recall of the classification task as a whole. All models suggested that the most critical feature is the time scheduled for the connection and all of them gave none to marginal importance to features such as age or gender.**

***Index Terms*—Flight Connections, Imbalanced Classification, Interpretable Models, Machine Learning, Model Explanation**

## I. Introduction

With the ever-increasing passenger demand, airports worldwide have to face traffic congestion problems at several levels, including, but not limited to, arrival and departure delays and bottlenecks within terminal facilities. Hence, with the development of the Intelligent Transportation System (ITS), on which large amounts of data are recorded every day, many approaches have been proposed to deal with the different congestion problems based on data collection from ITS. The problems airports face are also of extreme importance for airlines that see the flight schedule as a critical factor for their profitability success and client satisfaction.

However, and this is not a problem specific to this domain, the approaches being developed nowadays tend to follow the widespread belief that the most accurate models must be inherently complicated and non-interpretable by humans. These are commonly called Black-Box models, and even the human entity in charge of modelling cannot understand what combination of variables the model is basing its predictions on. The result is a model with no cap on complexity.

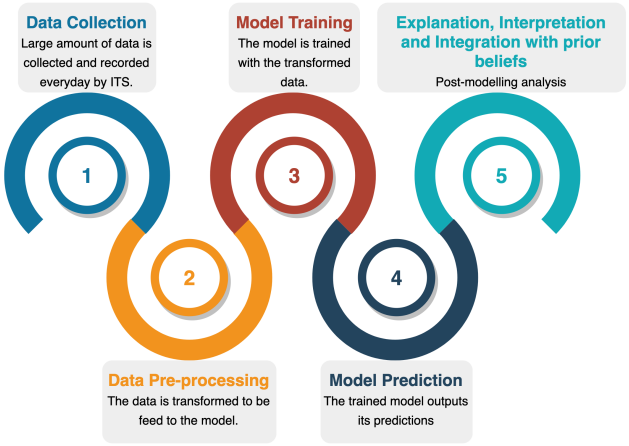



Fig. 1. Model development pipeline. The majority of approaches only focus on the first four steps, but this work places an important focus on the last step of the pipeline, and tries to understand what are the reasons for the model prediction.

The use of such models can be especially harmful when dealing with decisions that have a direct impact on people's life. As stated in [1] studies have shown that complicated Black-Box models used to predict the likelihood being arrested in the future are not more accurate than simple predictive models. Moreover, [2] showed that rule lists with a certificate of optimality could be as accurate as a state-of-the-art proprietary risk prediction tool, but that is entirely interpretable.

Conversely, White-Box models are models that try to answer the same type of questions as Black-Box models do but are fundamentally different from them in the sense that the former are mannered so that they provide reasoning on how the algorithm reached its predictions, something that did not happen with the latter. Interpretable models are often comprised of simpler models. However, most of them are not designed with interpretability issues in mind, they are just designed, like all other ML algorithms, to be as accurate as possible, and their interpretability is just an afterthought.[1]

The interpretability of ML models can be further branched according to several criteria. One of those criteria is the mechanism with which the interpretability is reached. Intrinsic and *post hoc* distinguishes if interpretability is achieved by capping the ML model complexity or by applying methods that analyze the model after training, respectively [3]. An example of intrinsic interpretability refers to models with a simple

structure and are therefore considered interpretable, such as shallow decision trees. In contrast, a *post hoc* interpretability model refers to applying explanation methods after the model training is concluded. *Post hoc* explanation methods can be applied to both White-Box models as well as Black-Box models. The definition of interpretability can also be branched in terms of the scope of the interpretability. Local or global if it explains a single instance, the model as a whole, respectively.

It is important to mention that despite the development of some algorithms capable of explaining Black-Box models predictions, this does not alleviate its inherent problems and the modeller confidence in the model's output still requires a high degree of abstraction. The solution to stop perpetuating bad practices is to design inherently interpretable models, as explained by Rudin in [4].

### A. Related Work

This paper addresses a conjugation of two problems and both of them have some literature on their specific broad topics. However, literature studying an interpretable approach to the prediction of missed connections is not yet available. In this section, we provide a wide but non-exhaustive review of works related to this paper.

Interpretability in the context of ML was defined in [5] using a unique framework called PDR — Predictive, Descriptive, Relevant — for discussing interpretations. This framework provides three prerequisites for evaluation: predictive accuracy, meaning the quality of a model's fit; descriptive accuracy, the degree to which an interpretation method captures the relationships learned by the model; and relevancy, if it provides insight for a particular audience into a chosen domain problem with relevancy primarily judged relative to a human being. Before this definition, Doshi-Velez and Kim [6] described interpretability in terms of the model's ability to elucidate in intelligible terms a human.

Although only recently gaining importance, Interpretable Machine Learning (IML) models have been around under-explored for many years [7]. Linear regression models were used as early as the 19th century and have since then grown into, for example, generalized additive models [8].

After the mid-2010s and even with the rapid development of deep learning models the research in the field of IML did not stop and many new IML methods have been proposed since. Many of those models are model-agnostic, but explanation techniques specific to deep learning and tree-based ensembles models were also subject to research. Nowadays, regression analysis and rule-based ML remain relevant topics and the linear regression model has seen many extensions proposed [9]. Rule-based ML research extensions have also been proposed. For example, these two domains are even blending as seen in model-based trees [10] or the RuleFit algorithm [11]. Both regression models and rule-based ML models serve as native ML algorithms, and as sub-blocks for many IML approaches.

In terms of the current research stage, the IML field has seen consolidation in terms of knowledge with, for example, [12] and work about defining interpretability like [13] or evaluation of IML methods [14].

Historically, the airline scheduling problem is separated into four more minor problems: Flight Scheduling Problem (FSP), Fleet Assignment Problem (FAP), Aircraft Maintenance Routing Problem (AMRP), and finally Crew Scheduling Problem (CSP), as stated in [15].

In past work, we can see that most of the attempts did not include an effort to cover all sub-problems of the big picture problem at once. The first problem that airlines typically need to solve before start operations is the FSP since the other sub-problems are dependent on this. The goal is to reach the end of this stage with a timetable containing a list of all flight schedules, while considering some constraints and limitations, just as presented by [16] that considered the influence of market constraints such as passenger demand and ticket price.

Still in the topic of FSP Yan and Young first presented in [17] a study on the sub-problem that considered the expectation of demand variation by passengers. This work was later improved in [18]. The difference between this second work and the original research was the consideration of the airline's market share. However, the two models failed to consider the variability and uncertainty of the market share percentage and the demand by passengers. To overcome these limitations, other studies like [19] and [20], investigated the variability of market share and passenger demand. These two works shifted towards a more authentic understanding of the airline industry since they consider market share fluctuations and passenger demand. Nonetheless, several other researchers worked on the flight scheduling problem and methods to increase its robustness. That was the case in [21].

Regarding the other dimensions of the overall problem, [22] presented a simple FAP solution by applying integer linear programming based on the structure of a connection network and then [23] and [24] improved upon the original model. These models were good starting points. However, the assumptions of fixed flight schedules and deterministic passenger demand compromise the applicability of these models in actual conditions.

When it comes to the other two pillars of the problem, the '90s saw researchers proposing solutions for the AMRP with a focus on the tactical side of the problem. The works of [25] and [26] are examples of such approaches. However, these approaches fail to consider some of the characteristics of the requirements of operational maintenance. Over the 2000s, research on the AMRP continued. However, it shifted and was dedicated to proposing models with a focus on the operational side of the problem, like seen in [27] that proposed an effective operational model. However, this model was limited in terms of the number of flights that it could handle. This limitation was then improved in [28] to handle a large number of flights. The last pillar, the CSP, moved from a daily horizon planning, like in [29], that despite being capable of handling large-scale problems and producing crew pairing to each specific day of the week had the limitation of assuming a daily repetition of all flights in order to simplify the computation. Since both

the daily repetition and the weekly repetition assume a fixed departure time, a stochastic and robust crew pairing approach was developed in [30] to balance this point.

One of the main drawbacks of past solutions and approach is that each stage, i.e, each sub-problem, is solved independently of other sub-problems, which means that the solution for one part of the problem as a whole, might not be optimal for the following steps. This motivated researchers to pursue models that integrate multiple sub-problems simultaneously. An example of such innovation was the introduction of integrated models to simultaneously solve both the FSP and the FAP at once, in [31].

Moreover, most approaches presented so far correspond to algorithms and mathematical solutions for their respective issues and have failed to work with real-world data. Studies using collected data tend to focus on the prediction of flight delays and ways to mitigate them. An example of that was the work introduced in [32], on which they proposed a new model for predicting air traffic delays using both temporal and network delay states with a 2h to 24h advance. [33] investigated the relationship between delays, delay propagation, and delay causes with aircraft, crew connections and passenger connections using a Bayesian Network in a delay-tree framework. The work in [34] proposes a novel analytical-econometric approach to understand delay propagation patterns and associated mitigation measures using flight data as the backbone of the analysis. To compliment the work in [34], [35] proposed a delay causality network and [36] investigated the mutual influence between the airline network structure and the airport congestion. Their study suggested that airlines with hub-and-spoke structures react less to delays than airlines operating fully connected networks.

The problem of predicting flight delays based on flight information only was the object of study in more recent research. For example, [37] presented a study analyzing data from an airport and presented a deep learning flight delay prediction model based on a multi-factor approach. The development of models with passenger data is generally more difficult due to the lack of available resources. However, [38] established relationships between several factors. These included, both passenger and flight delays, and passenger cancellation rates and load factors. [39] developed multiple ML frameworks centred around passenger data that additionally did a *post hoc* explanation analysis for the Black-Box model predictions. However, all those frameworks used the same Black-Box model, XGBoost, and did not try to include intrinsically interpretable models.

This paper contributions is a novel approach to the determine whether a passenger will have a successful connection or not. This approach uses the same data structure and content across multiple algorithms and:

1) develops two sets of distinctive predictive models belonging to the interpretable and uninterpretable categories;
2) compares the models performance and costs to understand what types of benefits come with the use of one model category over the other;
3) identifies what type of feature combinations the models are basing their predictions on.

## II. Data Preprocess

For the construction of the passenger-centric models developed in this paper, the data consists of 3 different datasets, all of which belong to TAP Air Portugal, and refer to the period between January 2019 and February 2020, encompassing 14 months. TAP is a legacy carrier based in Portugal that, like most of the other legacy carriers still operates the Hub-and-Spoke model where nearly all connecting passengers traveling with TAP pass through Lisbon airport on-route to their finally destination.

All 3 datasets contained distinct, however, complementary data. Serviço de Estrangeiros e Fronteiras, known as SEF, is the force responsible for the control of the border in Portugal and the first dataset was relatively simple, with 8815 rows and 3 columns with information stating whether the connecting passenger should go through the immigration office or not. This dataset depicted all possible combinations between departure and arrival airports. Then the hub dataset compiles information regarding 109 attributes of all flight movements (both arrivals and departures) encompassing 345,035 entries. These attributes include the time of arrival/departure, gate used, type of aircraft, etc. Finally, the passenger dataset included information regarding all 5,034,222 connecting passengers that had at least one leg of their flight operated by TAP Air Portugal. In total, 21 features were known and contained information such as arrival and departure flight numbers, gender, age, state of the connection, etc.

As the final goal is to have a single data set containing all relevant information with predictive power to build classification models, we performed an Exploratory Data Analysis (EDA), simultaneously with data cleaning process.

Based on all data provided and some exploration, one did some feature engineering in order to extract the most valuable information while keeping the total number of features as low as possible, maintaining interpretability. This process is helpful because it allows for the capture of information that was not explicit in the original data. The new features were: Scheduled Connection Time, Traffic Network and Traveling Class. All of these are self explanatory except for the Traffic Network which is a relevant feature in the context of an airline that operates flights within the European Union single air space as this acts under a single legislation. TAP's connecting passengers can therefore be traveling between two airports belonging to the Schengen space, meaning that they won't need to go through the immigration office, between two non-Schengen airports which implies going through security and the immigration office or between a Schengen and a non-Schengen airport (or vice-versa) which will also require some type of immigration check.

The engineered features were added to some other previously selected adding up to 11 in total. These features include

a lot of valuable information largely due to the immense information contained in the engineered features. The remaining features are: incoming and outgoing flight designator, sex, age, type of traveler (solo or group), day of the week (Sunday through Saturday) and day of the month. The obtained dataset included a lot of data, however, some of that was either out of the scope of the current work or was unusable. To start, one removed all rows corresponding to incoherent connections and/or non-TAP flights since they are not within the scope of this paper. We also removed all rows that contained at least one missing value.

After applying all these transformations, the dataset consists of 3,451,979 samples including 3,261,690 successful connections and 190,289 unsuccessful connections. As a next step one performed a stratified label-based train/test/validation split and the resulting data sets corresponded to 80%, 10% and 10% of the global data, respectively. In more concrete terms, the train data set has 2,761,583 samples and both the validation and test sets have 345,198 samples. The proportion of samples was kept during the process and it corresponded to around 94.5% of the samples being on the successful connections class and the remaining 5.5% on the unsuccessful connections class, making this an imbalanced problem.

The final steps involved the encoding and the scaling processes of the features. The choice of encoding technique for all features that required it was to perform a target encoding in order to keep the number of features as low as possible. To avoid overflow we created a new dataset with all 11 features were scaled used in the DNN and Logistic Regression algorithms.

## III. Chosen Approach

The goal of the Decision-Making Model being designed is to ascertain whether a connecting passenger is likely or not to miss the second leg of the journey. With that in mind, the information available shall be the same as the information known by the airline after the flight schedule is made public and the passengers booked their flight, but still prior to the flight date.

The aim of this model is to understand how the airline can translate data about the passengers into valuable insights. The data includes the features already discussed in the previous section and the airline should, after carefully analyzing the models, be able to notice trends and clusters and to understand what profile of passengers is most at risk of missing the connection and will do it knowing the relative importance of each feature for the final prediction.

### A. Imbalance

As one is dealing with an imbalanced dataset, an initial step is to rebalance the training data so that the data fed to the different classification algorithms is not biased towards the majority class. With this objective in mind, one will oversample the minority class using a probabilistic approach that learns the distribution of the class as a sum of Gaussian distributions, known as Gaussian Mixture Models (GMM).

To generate new samples, the first step is to select the relevant elements such as the number of Gaussian distributions ($K$), the mean ($\mu_k$) and co-variance ($\Sigma_k$) of each of those distributions.

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) \tag{1}$$

The model in 1 is known to be capable to approximate any distribution as long as the number of models, $K$, is large enough. [40]

To select the relevant parameters for the GMM, one used two important statistical models which attempt to quantify the performance on the training and the complexity of the model. The AIC and BIC are defined by:

$$\begin{aligned} AIC &= -2log(L) + 2K \\ BIC &= -2log(L) + 2Klog(N) \end{aligned} \tag{2}$$

where L is the likelihood, K the number of parameters, and N the number of samples. The AIC prioritizes model performance over model complexity, which might result in the selection of more complex models, whereas the BIC penalizes greatly model complexity, meaning that more complex models are less likely to be selected. To both of them, the lower the value the better the model is.

### B. Baseline

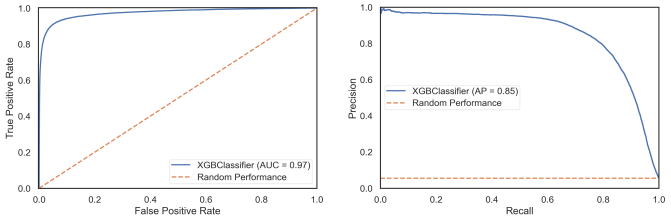
As the baseline classifier, one considered the system used by the airline that assumes as the criterion the Minimum Connecting Time (MCT) established by ANA - Aeroportos de Portugal, which is defined as 60 minutes. This means that the airline assumes that a passenger will miss the connection if the connection time is below 60 minutes.

The baseline model has an accuracy of 0.92 and the $AUC_{ROC}$ is equal to 0.61 and the $AUC_{PR}$ to 0.28. The model has a good performance on the majority class correctly labeling 95% of instances, however, the performance on the minority class is lower and the model miss-labeled around 72% of instances.

## IV. Modeling

Before feeding the training data and starting the modeling stage, one implemented the oversampling sampling technique described in the previous section.

The number of newly created samples corresponds to the difference between the total number of samples of the majority class, 2,609,352 and the number of samples of the minority class, 152,231 which translates to 2,457,121 new artificial samples. The new samples were generated after fitting the GMM model to the data corresponding to the minority class, using 1000 components and a diagonal covariance matrix since this was the combination of parameters that yielded the best results both in terms of AIC as in terms of BIC.




(a) ROC-AUC curve of the XGBoost algorithm.

(b) PR curve of the XGBoost algorithm.

Fig. 2. XGBoost model results.

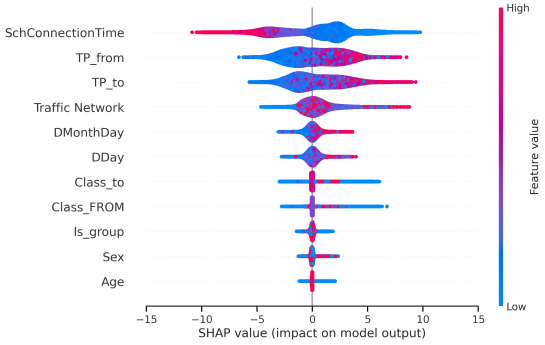



Fig. 3. SHAP summary plot for the XGBoost algorithm.

### A. *Black-Box Models*

In total, one trained 2 Black-Box models. The first was XGBoost, a Tree Ensemble Machine Learning algorithm that makes use of the gradient boosting framework and has a decision tree basis. Recent results suggest that in regard to structured data (also known as tabular data), decision tree based algorithms are considered to outperform the state-of-the-art black-box models, artificial neural networks. It can be used to solve a variety of different problems such as regression, classification, ranking, or even some user-defined prediction problem.

The second model was an artificial neural network, more specifically a Deep Neural Network. These are structures inspired by biology, that learn how to perform tasks by studying previous examples and are formed by a set of connected neurons. Information is transmitted between connected neurons in both directions that are typically organized in layers and different layers perform different transformations on their inputs.

*1) XGBoost:* After the selection of the hyperparameters, the final model was trained using the whole data and the selected combination of hyperparameters.

The XGBoost algorithm obtains the final predictions through weighting the results from all ensemble learners since this is a boosting algorithm. The output from the model assigns, to each instance being evaluated, its probability of belonging to the first class (successful connection) and the probability of belonging to the second class (unsuccessful connection). These two values add up to 1 in all instances.

Figure 2 shows the plot for the ROC-AUC and PR curves for both the specified thresholds. The $AUC_{ROC}$ is equal to 0.97, the Average Precision (AP) to 0.85 and the $AUC_{PR}$ to 0.80. As a whole, the accuracy is 0.98. Looking more concretely at the predictions, one can see that the model correctly predicted 98.9% of the majority class instances for the default threshold value, missing only 1.1% of the unsuccessful connection instances. However, the performance on the unsuccessful connection class was lower, and the model predicted the correct label on 79.1% of the instances and missed the label 20.9% of the time.

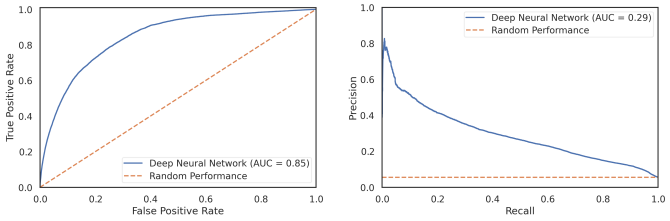
An important part of the scope of this paper is to go beyond the model output and enter the domain of the explainability and interpretability. The way that one chose to deal with this component of the post-analysis is to use the principles stated in Section I-A and make use of the SHAP framework that provides both local and global interpretations. The following analysis will be focused on more general interpretations that account for the effect of all instances used to train the model.

The SHAP summary plot accounts not only for the feature importance but also its effects and allows for a more thorough analysis of the specificity of each of the features. This representation is depicted in Figure 3. Each instance will be represented by the correspondent Shapley value on the summary plot. The y-axis is reserved for the features which are order by their relative importance and the height of the clusters indicate the distribution of the Shapley values per feature. The x-axis represents the Shapley value. The color attributed to each point represents the value of the instance on the corresponding feature, ordered from low to high.

The feature with the highest importance is the Schedule Connection Time and, as shown in Figure 3, samples with low connection time are associated with positive SHAP values, which means that a low connection time value contributes to classify the connection as unsuccessful. This is aligned with prior knowledge, since it is consistent with common sense. From the distribution of the instances throughout the feature span, it can also be seen that some low and medium values for the Schedule Connection Time still have a negative impact and contribute to classify as a successful connection however most samples are located within the range from 0 to 4, contributing to the classification as unsuccessful. On this feature, there is a clear separation between low values that contribute positively, and high to medium values that contribute negatively.

The next two most importance features, the incoming flight and the outbound flight number, despite assuming an important role in the prediction, cannot be completely analyzed since the data is encoded and represents a categorical feature and there is no ordinal logic behind it.

Then, the traffic network variable has a mixed impact on the final prediction. For high values, it can either have a high positive impact (classify the connection as unsuccessful) or a more moderate one. For low values it can either have a positive impact, however, not as high as the previous statement, or have a negative impact (classify the connection as successful). The




(a) ROC-AUC curve of the DNN algorithm. (b) PR curve of the DNN algorithm.

Fig. 4. DNN model results.

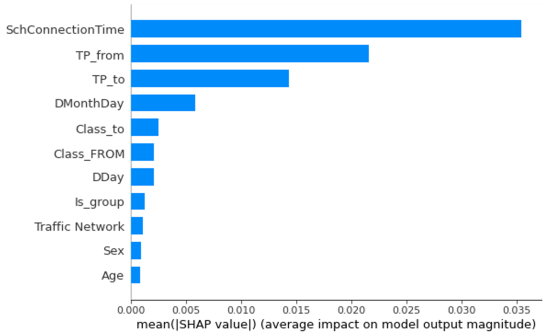



Fig. 5. SHAP feature importance for the DNN algorithm.

different combinations of transiting routes, i.e., the different combinations in terms of origin/destination airport pairs were encoded following the order Non-Schengen to Non-Schengen (NN) < Non-Schengen to Schengen (NS) < Schengen to Non-Schengen (SN) < Schengen to Schengen (SS). Non-Schengen airports are located outside the Schengen area, which is a supra-national agreement that includes most of the European Union plus some bordering countries. It is interesting to notice that SS connections can have such different impacts on the final prediction and that most NN connections have a negative impact (classify the connection as successful) which perhaps comes from the fact that these type of connections are related with long-haul flights which tend to have more sparse schedules and higher connection times.

The next feature, DMonthDay, has a scattered distribution which come naturally from the its meaning since it represents the encoded day of the month (from 1 until 31) when the connection took place. Due to the moving nature of the placement of weekends and different weekdays though as numbers, it is difficult to extract information on this behavior, however, this feature holds some relevant information since SHAP values are not zero. Instances with low values for the day of the week feature, DDay, tend to contribute for the classification of the instances as successful and higher values in this feature tend to contribute to classify them as unsuccessful.

For the traveling class, both inbound and outbound, the order on which the different categories were encoded is the same and is as follows: Business < Groups < Economy < Allots < R1. For the case of the outbound traveling class, low values have a non-negligible impact on the final predictions, which means that Business class instances can either have a high positive impact or a modest negative impact. Whereas in the case of the incoming traveling class, the business class keeps the same behavior as in the incoming class feature.

The final three features have the least impact on the predictions. Nevertheless, some general trends can be seen on those features. To start with, the three have most of the instances concentrated around zero, which indicates a null impact. Then for both the traveling within a group and gender features, it is possible to spot a shy partition of the data on which high values tend to have a positive impact and low values a negative impact. Both features are binary, which means that passengers traveling in group tend to have a slight positive impact (classify the connection as unsuccessful), whereas passengers traveling alone tend to classify the connection as successful, i.e., have a negative impact. When considering the gender of the passengers the impact is even less expressive than in the case of the group feature and men tend to have a positive impact contrary to what happens with a woman that has a negative impact. The last binary feature, the Age, has no impact for high values (adults) and moderate positive or negative impact for low values (infants).

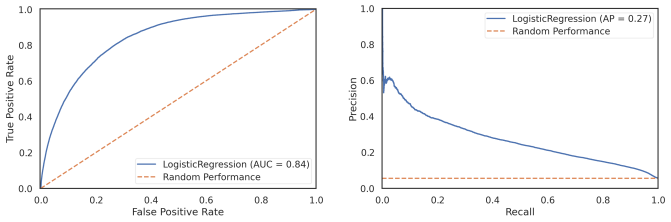
*2) Deep Neural Networks:* The procedure after the selection of the hyperparameters for the DNN is the same as the one followed when training the XGBoost.

The whole data was fed to the architecture designed based on the chosen hyperparameters. The DNN final predictions include a single value in the range from 0 to 1.

Figure 4 shows the plot for the ROC-AUC and PR curves. The $AUC_{ROC}$ is equal to 0.85 and an the $AUC_{PR}$ is equal to 0.29. Overall the results of this Black-Box model are not as good as the ones seen in XGBoost. Although very popular and the state-of-the-art for some machine learning applications Neural Networks don't perform well in structured imbalance data as they perform in unstructured like images. Overall, one can say that the model performance follows the same trend seen with the XGBoost algorithm, and the performance on the majority class surpasses the model's performance on the minority class of the test dataset. With the DNN the results on the majority class were below but close to the results seen with the XGBoost; however the results on the minority class were well below the ones seen with the XGBoost making the macro averaged metrics of the DNN algorithm between 0.64 and 0.66 contrary to the range from 0.87 to 0.89 seen in the XGBoost. As a whole, the accuracy in the test dataset is 0.92.

Applying the default threshold of 0.5 the model performed fairly well in the majority class, missing only around 5% of the predictions. However the case with the minority class is different and the model can only predict 36% of the instances correctly, miss-labeling the remaining ones.

Figure 5 shows the SHAP feature importance for the DNN model. It is easily seen that the feature that has the highest impact in the final prediction is the Schedule Connection Time following the behavior of the XGBoost model. The next two




(a) ROC-AUC curve of the Logistic Regression algorithm.

(b) PR curve of the Logistic Regression algorithm.

Fig. 6. Logistic Regression model results.

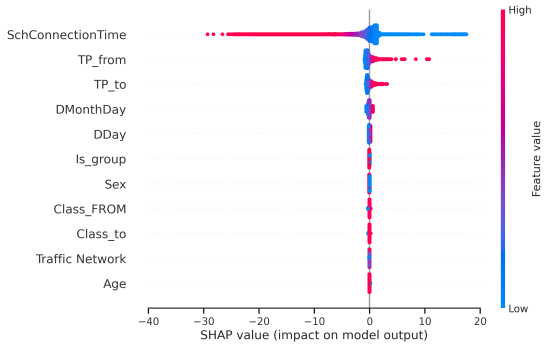



Fig. 7. SHAP summary plot for the Logistic Regression algorithm.

most important features also follow the trend seen in the XGBoost interpretation and are the incoming and outgoing flight codes, respectively. Here, and imitating the behavior seen previously, the first three features account for the majority of the interpretative value of the predictions made by the model. Starting from the 4th most impactful feature, it is possible to spot a clear divergence from the previous model, and the remaining features, except for the day of the month, assume a similar yet marginal contribution for the model's predictions. However, the traveling class (both inbound and outbound) and day of the week have a higher impact than the other three. The overall order is also similar to XGBoost's explanations, with age and gender being ranked at the bottom again.

### *B. White-Box Models*

In total, one trained two white-box models and did a preliminary study on a third. The first trained was a simple logistic regression, which is one of the simplest models available for binary classification problems. The logistic regression algorithm models the probabilities for classification problems with two possible outcomes and it can be seen as an extension of the linear regression model but applied to classification.

The second model to be trained was a tree based model that split the data according to certain cutoff values in the features. Different subsets of instances of the dataset are created through the splitting process and the final subsets are called terminal or leaf nodes and the intermediate subsets are called internal nodes or split nodes.

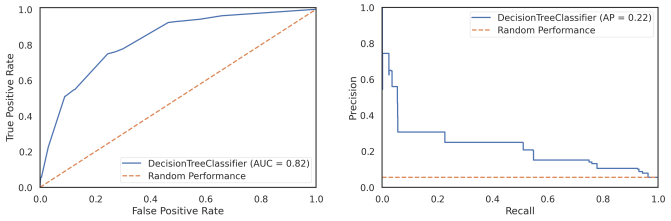
As an extra attempt to build another interpretable model one studied the RuleFit method. This a rule-based algorithm that learns a sparse linear model with a composition of the original features together with a number of new features that are decision rules. These new features capture interactions between the original features. This algorithm is primarily used for regression tasks, however one decided to evaluate its performance on this classification task.

*1) Logistic Regression:* The whole data was fed to the model which outputs, to each instance being evaluated, its probability of belonging to the first class (successful connection) and the probability of belonging to the second class (unsuccessful connection).

As this is a very simple White-Box model, the results are not as good as the ones found with DNN or especially with XGBoost. However, the performance in the minority class is comparable to the found with the DNN algorithm which is remarkable. Figure 6 shows the plot for the ROC-AUC and PR curves. The $AUC_{ROC}$ is equal to 0.84, which is close to the value obtained with the DNN, and an the $AUC_{PR}$ is equal to 0.46, above the value obtained with the DNN. We can say that the model performance follows the same trend seen with the Black-Box models, and the performance on the majority class surpasses the model's performance on the minority class of the test dataset, except for the Recall. However, the difference in performance between the two classes on the Precision metric is even greater than the difference seen in the former models, and the F1-score is close to the difference seen in the DNN. In terms of macro-averaged metrics, this model performance is worse than the XGBoost for all assessed metrics. When comparing with the DNN, the Precision and F1-score macro-averages are lower yet similar, but the Recall is higher and equal to the value seen in the XGBoost. As a whole, the accuracy is 0.77.

Looking at the output from SHAP explanations only 3 features have an effective impact on the model's predictions. From those the feature with the highest importance is the Schedule Connection Time, and as shown in Figure 7 samples with low connection time are associated to positive SHAP values, which means that a low connection time value contributes to classify the connection as unsuccessful. This behavior is the same as the one observed with the interpretation of the XGBoost algorithm. From the distribution of the instances points throughout the feature span, it can also be seen that some low and medium values for the Schedule Connection Time have a negative or marginal impact classifying the instance as a successful connection. On this feature, there is a clear separation between low values that contribute positively, and low to medium values that contribute negatively.

*2) Decision Tree Classifier:* Before training the Decision Tree Classifier, one gave careful consideration to the selection of hyperparameters since this model's intrinsic interpretable nature might be lost if the tree grows in depth beyond reasonable. As stated in Section I the width of the tree grows exponentially with the depth so this is a very important parameter to tune. After the selection of the hyperparameters, the final model was trained using the whole data and the




(a) ROC-AUC curve of the Decision Tree algorithm.

(b) PR curve of the Decision Tree algorithm.

Fig. 8. Decision Tree model results.

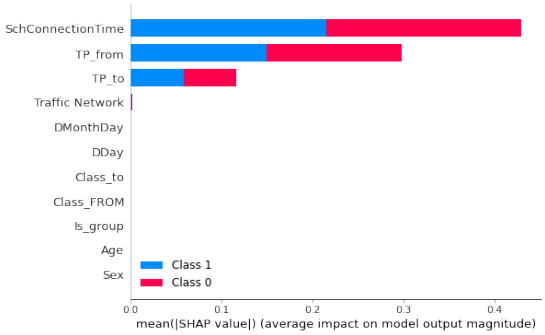



Fig. 9. SHAP feature importance for the Decision Tree Classifier algorithm divided by class.

selected combination of hyperparameters.

During the tuning one noticed that a depth of 5 corresponded to a good trade-off in terms of model performance while maintaining interpretability. Figure 8 shows the plot for the ROC-AUC and PR curves. The $AUC_{ROC}$ is equal to 0.82, the smallest value among all four algorithms, and an the $AUC_{PR}$ is equal to 0.46, the same value as the Logistic Regression. Once again, the model performance follows the same trend seen with the Black-Box models and with the Logistic Regression, and the performance on the majority class is better than the model's performance on the minority class except for the Recall that has similar values on both classes. The difference in performance between the two classes among the remaining metrics is similar to the one seen in the Logistic Regression model. The macro-averages scores are close to the ones seen with the Logistic Regression, and therefore lower yet similar to the ones seen with the DNN except for the Recall, which is higher in the interpretable models. As a whole, the accuracy is 0.75.

After training the model and building the decision tree, the results and their intrinsic interpretability are clear. The first split occurs by analyzing if the Schedule Connection Time is less than or equal to 126.5 minutes. This means that with this model the Schedule Connection Time is again the most relevant feature. After the initial split, each sub-branch is further split based on the incoming flight number code, although the criteria for the split are different in each sub-branch. The third and fourth splits are again based on either the Schedule Connection Time or the Incoming flight code, proving that these two features are the absolute most relevant for the classification by this model. Only in the 5th split, new features, namely the traffic network and the outgoing flight code, become relevant. The results from the analysis of the shape of the tree are coherent with both the feature importance method native to the algorithm and the SHAP summary, Figure 9. The figure also shows that all features had similar impact in both classes.

*3) RuleFit:* The entirety of the dataset was fed to the algorithm, however the results weren't promising. Keeping in mind that this is an intrinsic interpretable model, the goal is to reach a good performance while keeping the number of rules and the maximum depth of the auxiliary trees to the bare minimum.

Unfortunately, this was not the case and the model performance was poor even when having great flexibility in terms of the number of rules. Even though the model was trained with the balanced dataset, it was incapable of having a good classification performance on both classes and ended up assigning the majority of instances to the same class. This meant that it had 99.6% of true negatives but only 11.2% of true positives, missing all the others instances belonging to the unsuccessful class even when using 248 rules which would have make this model already uninterpretable.

## V. Costs & Results Assessment

### A. Costs

The analysis presented in this Section is based on the work developed in [39]. The original work presents the cost analysis of which we replicate the equations and approximations. All four strategies presented had different results than the results obtained by TAP baseline model. The costs associated with each type of predicted connection are difficult to breakdown since ultimately all passengers are different and have different needs. To overcome that, the original research found the relationship between costs that would make the model economically viable for the airline.

According to the original work, the airline will incur in 2 types of expenses: Precautionary Costs, $C_{Pre}$ and Corrective Costs, $C_{Cor}$. The current system in place, using the Baseline model, does not consider Precautionary costs and assumes a corrective only approach. When the model correctly predicts a True Positive (TP), the airline will have useful information to try to minimize the disruption and will incur in precautionary costs. Some possible actions include, but are not limited to, the delaying of the 2nd leg of the journey, assigning some airport personnel to escort the passengers to their next flight or even assigning a seat closer to the exit door and give those passengers priority. When the model predicts a False Positive (FP) the airline will incur in the same type of precautionary costs that we have seen in the TP case. And whenever the model predicts a False Negative (FN), the airline will incur in unexpected corrective costs to solve the missed connection. These costs may include assigning the passenger onto a new flight and making the arrangements necessary for the extended

TABLE I
COST STRUCTURE OF THE DIFFERENT APPROACHES.

| | Basline Model | Developed Frameworks |
|---|---|---|
| # Precautionary actions | No Actions | TP & FP |
| # Corrective actions | TP & FN | FN |

TABLE II
RELATIONSHIP BETWEEN PRECAUTIONARY AND CORRECTIVE COSTS FOR EACH MODEL.

| | $p_{min}$ |
|---|---|
| XGBoost | 1.25 |
| DNN | 3.14 |
| Logistic Regression | 6.14 |
| Decision Tree Classifier | 6.60 |

layover, which might include overnight accommodation or meal vouchers. Table I summarizes the costs.

The following step, performed in the original research, was to consider the difference in costs between the implementation *versus* no implementation of the XGBoost model. We followed the same approach but in the case of the current scope the difference will be computed to each one of the models independently. The authors derived Equation (3) to find how many times $C_{Cor}$ must be larger than $C_{Pre}$, $C_{Cor} = pC_{Pre}$, to make the model attractive. Since the goal is to lower the costs, the authors aimed at a negative change of costs, $\Delta C$,

$$\Delta C < 0 \Leftrightarrow p > \frac{FP + TP}{TP} \tag{3}$$

The results from Table II indicate that the solution outputted by each one of the models is worth pursuing if $C_{Cor}$ are at least $p$ times greater than $C_{Pre}$. The XGBoost had the best results in terms of the easiness of possible savings for the airline since it only requires $C_{Cor}$ to be 1.25 times greater than $C_{Pre}$. The remaining algorithms had worse performance, but the use of the Logistic Regression or Decision Tree Classifier might still be justifiable due to their interpretability and the importance the airline attributes to that characteristic.

### B. Results

All models were trained in similar conditions and assessed on the same test data, therefore a qualitative and quantitative comparison is possible. In terms of accuracy, the best model was, by far, the XGBoost with an accuracy score of 0.98. The DDN had a 0.92 accuracy followed by the Logistic Regression and the Decision Tree Classifier with 0.77 and 0.75, respectively. This shows that the Black-Box models had a higher accuracy than the White-Box models in this classification task.

However, given the fact that this is a highly imbalanced classification problem, the accuracy is not a good metric because the algorithm might be biased towards the dominant class. Therefore, looking at either the Precision, Recall or F1-score macro-average values or at the ROC-AUC or PR-AUC scores is a better indicator of the model performance. The quality of the model also depends on the optimal point between Precision and Recall. This trade-off depends on the severity of the issue in hands, in this case the costs incurred by the airline. Nonetheless, for cases with missed connections, it is advised to minimize the risk of not alerting the airline about a person that may be at risk of missing the connection by minimizing the Miss Rate or False Negative Rate:

$$FNR = \frac{FN}{P} = 1 - Recall \tag{4}$$

Since the model is predicting if a passenger will make a successful connection or not, the aim of the model is to have a high Recall value, meaning that a smaller number of FN will be predicted by the model. Looking at the Recall in the minority class, the White-Box models achieve a good performance and have the best score *ex aequo* with the XGBoost model. The results were as follows: XGBoost, Logistic Regression and Decision Tree Classifier with 0.75, the DNN with 0.36. However the results on the macro-averaged metric were not as good meaning that White-Box models do not perform extremely well on the majority class but were still better than the DNN.

Regarding the models interpretability/explainability, all approaches presented different results. In the case of the XGBoost model, the Black-Box tree ensemble algorithm, the explanation results given by SHAP differ from the feature importance results given by the built-in tool. In those cases, not only the less importance features saw a change in the rank but also among the top ranks the two approaches showed different results.

In the second Black-Box model, the DNN, the explanation given by SHAP was mostly coherent with the explanation given by SHAP on the XGBoost algorithm predictions. This was the only model among the 4 that did not have any form of built-in feature importance tool.

In the case of the White-Box models, the results from the models intrinsic interpretability were mostly coherent with the explanations by SHAP. The Decision Tree based its predictions in 4 features, which are the same features to which SHAP outputted any level of importance and the Logistic Regression coefficients, when ordered by their absolute value correspond to the order of feature importance outputted by SHAP. Only a slight deviation in the behavior was noticed in terms of the DMonthDay feature to which SHAP outputted an impact bellow the incoming flight number although the 2 features coefficients are not that different.

## VI. CONCLUSION

The major achievement of the present work was the introduction of model interpretability in the domain. The airline industry is a multi-billion dollar industry with razor thin profit margins. This means that all improvements in the business are welcomed but at the same time decision makers might be unwilling to make decisions based solely in the output of a Black-Box model with no knowledge of the reasoning behind it whatsoever. The model and explanations developed allow for the prediction of the likelihood of missed connections

which might result in capital savings to the airline if it acts preventively, and do this while presenting the reasons for such decisions allowing for the ownership by responsible entities.

## REFERENCES


[1] C. Rudin and J. Radin, "Why are we using black box models in ai when we don't need to? a lesson from an explainable ai competition," *Harvard Data Science Review*, vol. 1, 10 2019.

[2] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin, "Learning certifiably optimal rule lists," *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017. [Online]. Available: https://doi.org/10.1145/3097983.3098047

[3] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/8/832

[4] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 10 2018. [Online]. Available: https://arxiv.org/abs/1811.10154v3

[5] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 44, p. 22071–22080, Oct 2019. [Online]. Available: http://dx.doi.org/10.1073/pnas.1900654116

[6] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, arXiv: 1702.08608.

[7] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable machine learning – a brief history, state-of-the-art and challenges," 2020, arXiv: 2010.09337.

[8] T. Hastie and R. Tibshirani, "Generalized Additive Models," *Statistical Science*, vol. 1, no. 3, pp. 297 – 310, 1986. [Online]. Available: https://doi.org/10.1214/ss/1177013604

[9] M. Fasiolo, R. Nedellec, Y. Goude, and S. N. Wood, "Scalable visualization methods for modern generalized additive models," *Journal of Computational and Graphical Statistics*, vol. 29, no. 1, pp. 78–86, 2020. [Online]. Available: https://doi.org/10.1080/10618600.2019.1629942

[10] A. Zeileis, T. Hothorn, and K. Hornik, "Model-based recursive partitioning," *Journal of Computational and Graphical Statistics*, vol. 17, no. 2, pp. 492–514, 2008. [Online]. Available: https://doi.org/10.1198/106186008X319331

[11] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, pp. 916–954, 2008.

[12] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[13] K. Sokol and P. Flach, "Explainability fact sheets," *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Jan 2020. [Online]. Available: http://dx.doi.org/10.1145/3351095.3372870

[14] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable ai systems," *ACM Trans. Interact. Intell. Syst. 1, 1, Article*, vol. 1, p. 46, 2020.

[15] A. E. Eltoukhy, F. T. Chan, and S. H. Chung, "Airline schedule planning: A review and future directions," *Industrial Management and Data Systems*, vol. 117, pp. 1201–1243, 2017.

[16] L. H. Lee, C. U. Lee, and Y. P. Tan, "A multi-objective genetic algorithm for robust flight scheduling using simulation," *European Journal of Operational Research*, vol. 177, pp. 1948–1968, 10 2007.

[17] S. Yan and H. F. Young, "A decision support framework for multi-fleet routing and multi-stop flight scheduling," *Transportation Research Part A: Policy and Practice*, vol. 30, pp. 379–398, 1996.

[18] S. Yan and C. H. Tseng, "A passenger demand model for airline flight scheduling and fleet routing," *Computers & Operations Research*, vol. 29, pp. 1559–1581, 10 2002.

[19] S. Yan, C. H. Tang, and M. C. Lee, "A flight scheduling model for taiwan airlines under market competitions," *Omega*, vol. 35, pp. 61–74, 10 2007.

[20] H. Jiang and C. Barnhart, "Dynamic airline scheduling," *Transportation Science*, vol. 43, no. 3, pp. 336–354, 2009. [Online]. Available: https://doi.org/10.1287/trsc.1090.0269

[21] S. Lan, J.-P. Clarke, and C. Barnhart, "Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions," *Transportation Science*, vol. 40, no. 1, pp. 15–28, 2006. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/trsc.1050.0134

[22] J. Abara, "Applying integer linear programming to the fleet assignment problem," *Interfaces*, vol. 19, no. 4, pp. 20–28, 1989. [Online]. Available: http://www.jstor.org/stable/25061245

[23] C. A. Hane, C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. Sigismondi, "The fleet assignment problem: Solving a large-scale integer program," *Mathematical Programming 1995 70:1*, vol. 70, pp. 211–232, 10 1995. [Online]. Available: https://link.springer.com/article/10.1007/BF01585938

[24] R. A. Rushmeier and S. A. Kontogiorgis, "Advances in the optimization of airline fleet assignment," *Transportation Science*, vol. 31, no. 2, pp. 159–169, 1997. [Online]. Available: https://doi.org/10.1287/trsc.31.2.159

[25] Z. Liang and W. A. Chaovalitwongse, "The aircraft maintenance routing problem," *Springer Optimization and Its Applications*, vol. 30, pp. 327–348, 2009.

[26] K. T. Talluri, "The four-day aircraft maintenance routing problem," *Transportation Science*, vol. 32, no. 1, pp. 43–53, 1998. [Online]. Available: https://doi.org/10.1287/trsc.32.1.43

[27] C. Sriram and A. Haghani, "An optimization model for aircraft maintenance scheduling and re-assignment," *Transportation Research Part A: Policy and Practice*, vol. 37, pp. 29–48, 10 2003.

[28] M. Başdere and Ümit Bilge, "Operational aircraft maintenance routing problem with remaining time consideration," *European Journal of Operational Research*, vol. 235, pp. 315–328, 10 2014.

[29] K. L. Hoffman and M. Padberg, "Solving airline crew scheduling problems by branch-and-cut," *Management Science*, vol. 39, pp. 657–682, 1993.

[30] I. Muter, I. Birbil, K. Bülbül, G. Şahin, H. Yenigün, D. Taş, and D. Tüzün, "Solving a robust airline crew pairing problem with column generation," *Computers & Operations Research*, vol. 40, pp. 815–830, 10 2013.

[31] M. Lohatepanont and C. Barnhart, "Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment," *Transportation Science*, vol. 38, pp. 19–32, 2004.

[32] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 231–241, 10 2014.

[33] C. L. Wu and K. Law, "Modelling the delay propagation effects of multiple resource connections in an airline network using a bayesian network model," *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 62–77, 10 2019.

[34] N. Kafle and B. Zou, "Modeling flight delay propagation: A new analytical-econometric approach," *Transportation Research Part B: Methodological*, vol. 93, pp. 520–542, 10 2016.

[35] Q. Li and R. Jing, "Characterization of delay propagation in the air traffic network," *Journal of Air Transport Management*, vol. 94, p. 102075, 10 2021.

[36] X. Fageda and R. Flores-Fillol, "Airport congestion and airline network structure," *Advances in Airline Economics*, vol. 6, pp. 335–359, 2017.

[37] 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, 10 2019.

[38] S. Bratu and C. Barnhart, "An analysis of passenger delays using flight operations and passenger booking data," *Air Traffic Control Quarterly*, vol. 13, no. 1, pp. 1–27, 2005. [Online]. Available: https://doi.org/10.2514/atcq.13.1.1

[39] M. Guimarães, "Predicting passenger connectivity in an airline's hub airport," Master's thesis, Instituto Superior Técnico – Universidade de Lisboa, Av. Rovisco Pais. 1049-001 Lisboa, 1 2021.

[40] H. W. Sorenson and D. L. Alspach, "Recursive bayesian estimation using gaussian sums," *Automatica*, vol. 7, pp. 465–479, 7 1971.