Modelling the Likelihood of Air Traffic Management Regulations due to Weather at Airports

Ramon Dalmau, Jonathan Attia & Gilles Gawinowski
EGSD/INO/ENG&SHO
EUROCONTROL Innovation Hub (EIH)
Brétigny-Sur-Orge, France

***Abstract*—Adverse weather conditions, such as low visibility, can have a significant impact on airport capacity. When the capacity reduction is substantial and traffic demand remains high, air traffic flow management regulations are implemented to ensure that traffic demand remains below the (reduced) capacity. Traditionally, regulations are established by human operators hours in advance, relying on their subjective perception of the weather forecast and expected traffic demand. This paper introduces a machine learning model explicitly designed to capture the likelihood of air traffic flow management regulations based on weather conditions and traffic demand. To address the inherent noise in the dataset labels, stemming from decisions made in advance by operators relying on uncertain data, confident learning techniques are proposed to build a more robust and reliable model. The robustness of the model against noise is further enhanced by enforcing monotonic constraints during the training process. The experiments demonstrate satisfactory model performance for major European airports that frequently encounter adverse weather conditions. The main objective of this model is to assist operators in determining the effectiveness of implementing regulations and aid airlines in predicting potential delays or airborne holdings resulting from adverse weather.**

***Keywords*—airport capacity; adverse weather; machine learning**

I. Introduction

When the expected traffic demand exceeds the capacity of the airport – which depends on the runway configuration and the weather conditions –, air traffic flow management (ATFM) regulations are frequently implemented to prevent overloads. Flights affected by ATFM regulations are assigned ground delays with the purpose of smoothing the traffic demand.

ATFM regulations are occasionally implemented at European airports. From the resurgence of air traffic following the lifting of COVID-19 pandemic restrictions on June 15th, 2021 until May 31st, 2023, a total of 1.3K ATFM regulations were implemented at airports within the European Civil Aviation Conference (ECAC) region, generating 430K minutes of delay. Notably, 13% of these ATFM regulations were caused by adverse weather. ATFM regulations caused by adverse weather are typically implemented well in advance, using predictions of traffic demand and weather forecasts to anticipate potential negative impacts on airport capacity. Needless to say, precise and accurate estimation of airport capacity is essential for ensuring the effective and efficient implementation of ATFM regulations. Overestimating airport capacity may force flights to wait in holding stacks, whereas underestimating airport capacity may result in an excessive ATFM delays.

In recent years, modelling and predicting the impact of weather on airport capacity has become a prominent area of research. For example, [1] utilised artificial neural networks (ANNs) to predict airport capacity considering weather conditions. [2] also utilised ANNs, but to classify airport performance into various categories. [3] quantified the influence of convective weather on terminal area capacity. Finally, in a recent study [4], the authors proposed a model to predict the *peak service rate*, a reasonable proxy of airport capacity, conditioned on the weather conditions and runway configuration. Predicting airport capacity can assist human operators in determining precise entry rates in the event of implementing ATFM regulations. This paper takes an additional step by modelling the likelihood of human operators implementing ATFM regulations due to adverse weather at the airport.

It is important to acknowledge that the likelihood of ATFM regulations due to weather for en-route airspace sectors has been addressed in the existing literature. For instance, [5] proposed regression and classification models to predict airspace performance characteristics, including entry count, the number of flights impacted by weather regulations, and the activation of regulations due to weather. Similarly, [6] presented a machine learning model for capturing the relationship between traffic demand, weather and the presence of ATFM regulations. To the best of our knowledge, the first attempt to predict ATFM regulations at airports was recently proposed by [7]. Their model, also built on ANNs, learned the presence or absence of ATFM regulations at the airport from historical observations.

As mentioned earlier, ATFM regulations due to weather are implemented with a certain look-ahead time, relying on a weather forecast to determine the expected capacity drop. In practice, however, the inaccuracies of weather forecasts [8] may lead to the actual period of capacity reduction differing from that predicted several hours ahead. To illustrate the impact of this problem when training machine learning models, let us consider a dataset comprising numerous observations. In this dataset, each observation corresponds to a specific time period, such as 1 hour, and includes a wide range of variables representing weather conditions, traffic demand, and the binary label indicating the presence (positive) or absence (negative) of ATFM regulation due to weather. At first glance, this dataset could be used to train a machine learning model to predict the likelihood of ATFM regulation due to weather, conditioned on the weather conditions and the traffic demand.

SESAR Innovation Days
27-30 November 2023
Seville

Some positive observations may be linked to regulations that were put into place due to pessimistic weather forecasts made several hours in advance. However, the actual weather conditions at the time of the observation might not have been as severe, rendering the regulation less effective than initially planned. On the other hand, some negative observations could be associated with periods where a regulation would have been beneficial in preventing airborne holdings, but was not implemented due to an overly optimistic weather forecast. By the time the actual weather conditions became clear, it was too late to put the regulation into effect. In essence, the dataset labels may contain noise, with some positive observations that should actually be negative, and vice versa. If this noise is not addressed, it could negatively impact the training of the model, leading it to learn incorrect or non-intuitive relationships between weather, traffic demand, and the likelihood regulation.

A similar problem was addressed in [9], where the authors attempted to learn the probability of flight diversion due to weather. In that case, some of the diversions observed in the dataset were attributed to other unpredictable reasons (e.g., medical emergencies) and should be considered as belonging to the negative class when the objective was to learn the mapping between adverse weather conditions and the probability of diversion due to weather. To address the noise in the labels, confident learning (CL), a method to automatically filter out likely mislabelled observations from the dataset, was adopted.

In this paper, CL is applied to filter out positive observations from the dataset that were regulated but likely no longer effective, as well as negative observations where an ATFM regulation was not implemented but could have been beneficial. Subsequently, a machine learning model is trained on the clean dataset to learn, with confidence, the relationship between adverse weather conditions, traffic demand, and the probability of ATFM regulation due to weather.

In the experiment, the potential of the proposed CL method is demonstrated by showcasing the performance of the model on a comprehensive two-year dataset comprising historical traffic and weather data from Europe's top 46 busiest airports.

II. Uncertainty Estimation in Dataset Labels

This section provides a summary of the content presented in [9] concerning uncertainty estimation in dataset labels using CL. For more comprehensive details about this method, readers are encouraged to refer to the original publication.

Let $[m] = \{0, 1, \dots, m-1\}$ denote the set of m class labels, and $\boldsymbol{X} := (\boldsymbol{x}, y)^n \in \left(\Re^d, [m]\right)^n$ the dataset composed of n observations $\boldsymbol{x} \in \Re^d$ with associated noisy labels $\tilde{y} \in [m]$. Our goal is to learn the mapping $\boldsymbol{x} \to y^*$ from the noisy observations in $\boldsymbol{X}$, being y^* the true (and unknown) label. Note that, in a binary classification problem, $[m] = \{0, 1\}$.

Whatever model is chosen, it can be abstracted as a parametric function, which unknown parameters $\boldsymbol{\theta}$ are adjusted during training to minimise the expected value of a loss function L:

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{|\boldsymbol{X}|} \sum_{(\boldsymbol{x},\tilde{y})\in\boldsymbol{X}} L\left(\tilde{y}, \boldsymbol{x}, \boldsymbol{\theta}\right). \tag{1}$$

For classification tasks, the categorical cross-entropy is typically used as loss function, which can be computed as:

$$L\left(i, \boldsymbol{x}, \boldsymbol{\theta}\right) = -\log \hat{p}\left(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta}\right), \tag{2}$$

where $\hat{p}\left(\tilde{y} = i; \boldsymbol{x}; \boldsymbol{\theta}\right)$ is the predicted probability of observation $\boldsymbol{x}$ belonging to class i, given the model parameters $\boldsymbol{\theta}$.

Training the model on the noisy dataset is likely to yield a set of parameters $\boldsymbol{\theta}$ that, in attempting to learn the unpredictable noise that minimises L, degrades the performance of the model on the real mapping $\boldsymbol{x} \to y^*$. Intuitively, the noisy observations should be ignored during the training process, and this is exactly what CL is designed to do.

CL builds on ideas that have been developed in the literature about noisy labels [10]–[15]. For a full coverage of theory and proofs, the reader is referred to [16]. The remainder of this section only summarises the fundamentals of CL that are required to judge the results and conclusions of this study. CL assumes that, before observing $\tilde{y}$, a class-conditional noise process (which is also unknown) maps $y^* \to \tilde{y}$, such that every label in class $j \in [m]$ may be independently mislabelled as class $i \in [m]$ with probability $p\left(\tilde{y} = i \,|\, y^* = j\right)$.

The first step in the CL process is to characterise the class-conditional label noise by estimating the joint distribution of noisy and true labels. This joint distribution is modelled as a $m \times m$-dimensional matrix $\boldsymbol{Q}$ that can be computed from:

1) the $n \times m$-dimensional matrix of out-of-sample predicted probabilities $\hat{\boldsymbol{P}}$, which rows correspond to observations and columns to class labels (e.g., $\hat{\boldsymbol{P}}_{i,j}$ is the probability of the i^{th} observation belonging to the class j), and
2) the n-dimensional vector of (observed) noisy labels.

where $\hat{\boldsymbol{P}}$ can be obtained via K-fold cross-validation (CV).

This process consists of splitting the full dataset into K disjoint subsets (also known as *folds*). Then, K independent copies of the model are trained. For each copy, one of the folds is held out for validation, and the other $K-1$ folds are used for training. Concatenating the predictions of the trained copies of the model on the corresponding validation sets yields the matrix of out-of-sample predicted probabilities $\hat{\boldsymbol{P}}$.

From the out-of-sample predicted probabilities and the noisy labels, it is possible to count the observations that are likely to belong to another class. The counts are captured by the confident joint matrix $\boldsymbol{C} \in \mathbb{Z}_{\geq 0}^{m \times m}$.The entry at the i^{th} row and j^{th} column of this matrix counts the number of observations labelled as class i with large enough $\hat{p}\left(\tilde{y} = j; \boldsymbol{x}, \boldsymbol{\theta}\right)$ to likely belong to the class j according to a per-class threshold t_k:

$$\boldsymbol{C}_{i,j} = \left| \left\{ (\boldsymbol{x}, \tilde{y}) \in \boldsymbol{X}_i : j = \underset{k \in [m']}{\arg\max}\, \hat{p}\left(\tilde{y} = k; \boldsymbol{x}, \boldsymbol{\theta}\right) \right\} \right|, \tag{3}$$

where $\boldsymbol{X}_i = \{(\boldsymbol{x}, \tilde{y}) \in \boldsymbol{X} \,:\, \tilde{y} = i\}$ is the subset of observations labelled as class i, i.e.:

$$[m'] = \left\{k \in [m] : \hat{p}\left(\tilde{y} = k; \boldsymbol{x}, \boldsymbol{\theta}\right) \geq t_k\right\}, \tag{4}$$

and the per-class threshold t_k is the expected (average) self-confidence of class k:

$$t_k = \frac{1}{|\boldsymbol{X}_k|} \sum_{(x,\tilde{y})\in \boldsymbol{X}_k} \hat{p}\left(\tilde{y} = k; \boldsymbol{x}, \boldsymbol{\theta}\right). \tag{5}$$

Then, the confident joint matrix $\boldsymbol{C}$ has to be calibrated so that row-sums match the observed marginals, i.e:

$$\bar{\boldsymbol{C}}_{i,j} = \frac{\boldsymbol{C}_{i,j}}{\sum_{j'\in[m]} \boldsymbol{C}_{i,j'}} |\boldsymbol{X}_i|. \tag{6}$$

Finally, the joint distribution of noisy and true labels can be estimated by normalising the calibrated confident joint matrix:

$$\hat{\boldsymbol{Q}}_{i,j} = \frac{\bar{\boldsymbol{C}}_{i,j}}{\sum_{i'\in[m]} \sum_{j'\in[m]} \bar{\boldsymbol{C}}_{i',j'}}. \tag{7}$$

Following the estimation of $\boldsymbol{Q}$, the next step is to identify the noisy observations and remove them from $\boldsymbol{X}$. Any rank and prune strategy can be adopted to identify noisy observations. The reader is referred to [16] for a comprehensive description of the main rank and prune strategies. In this paper, all observations counted in the off-diagonals of $\boldsymbol{C}$ have been considered as noisy. [16] demonstrated that this strategy (named confident learning) has attractive mathematical properties and yields excellent results in many applications.

After removing the noisy observations, the last step consists of training a copy of the model with the clean dataset. During training, one must account for missing data by weighting each observation in the loss function Eq. (2) according to the corresponding per-class weight $\omega_i = \sum_{j\in[m]} \hat{\boldsymbol{Q}}_{j,i} / \hat{\boldsymbol{Q}}_{i,i}$, $i \in [m]$:

$$L\left(i, \boldsymbol{x}, \boldsymbol{\theta}\right) = -\omega_i \log \hat{p}\left(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta}\right). \tag{8}$$

As with any machine learning problem, the train and test sets must be handled with caution in order to avoid information leakage. [16] suggested to clean the train and test sets independently when using CL. Specifically, the matrix $\hat{\boldsymbol{P}}$ for determining the matrix $\hat{\boldsymbol{Q}}$ of the train set should be computed by using the K-fold CV method as explained above. For the test set, $\hat{\boldsymbol{P}}$ should be generated from the probabilities predicted by a copy of the classifier trained on the entire noisy train set.

III. Experiment

In this paper, CL was employed to train a robust machine learning model for capturing the likelihood of ATFM regulation due to weather. This section outlines the experimental setup, with Section III-A presenting the various data sources, Section III-B providing details about the noisy dataset $\boldsymbol{X}$, and Section III-C presenting the machine learning model.

A. Data sources

The dataset was created by merging different types of data, including flight schedules, ATFM regulations, and meteorological reports. The rest of this subsection will provide further information about their sources and/or the processing methods.

1) Traffic demand: The EUROCONTROL's Aviation Intelligence Unit (AIU) kindly provided airport operator data flow (APDF) to compute the scheduled traffic demand (i.e., the scheduled number of hourly arrivals and departures). The APDF is established for 90 airports (as of April 2020) and includes extensive data for every flight, such as the scheduled and actual time of a movement (take-off time for departures and landing time for arrivals), the type of movement (arrival or departure), and the runway used. The APDF is provided monthly by the airport operators and integrated into a common database after undergoing data quality checks.

2) ATFM regulations: The start and end times of each regulation triggered by adverse weather and applied at any of the airports considered in the experiment during the analysed time period were extracted from EUROCONTROL's archive.

3) Meteorological reports: Meteorological aerodrome reports (METARs) from SADIS were used as weather observations. The METARs were processed using `metafora`[1], an open-source tool designed to transform textual meteorological reports into a vector representation including numerical (e.g., visibility, ceiling) and categorical (e.g., presence of thunderstorms or snow) features suitable for many machine learning models, particularly those based on decision trees.

B. Dataset

The dataset considers the Europe's top 46 busiest airports during 2022 and covers the period from June 15$^{\text{th}}$, 2021 to May 31$^{\text{st}}$, 2023. Each observation in the dataset corresponds to a 1-hour time window, spanning from 7AM to 10PM local time. These time windows start 15 minutes after the previous one. Each observation contains a vector of input features (or predictors), $\boldsymbol{x}$, along with the corresponding target value, y.

For the train-test split, the roughly 2M observations were randomly assigned, with 1.6M (80%) allocated to the train set and 397K (20%) to the test set. However, to avoid information leakage from the train set into the test set, a constraint was implemented to guarantee that all observations pertaining to the same airport and date (e.g., Zurich airport on June 15$^{\text{th}}$, 2021) are exclusively assigned to either the train or test set.

Table I provides a basic yet revealing description of the categorical and numerical features that compose $\boldsymbol{x}$ present in both the train and test sets. For the categorical features, the table includes the percentage of missing values, the number of unique categories, the most frequent (top) category, and the frequency of the top category. As for numerical features, the table displays the percentage of missing values along with the 5$^{\text{th}}$, Median, and 95$^{\text{th}}$ percentiles. The target value of each observation, y, represents the presence (positive) or absence (negative) of ATFM regulation due to weather during the corresponding time window. The number of positive observations in the train and test sets is 20.3K and 5K, respectively, representing a low proportion (1.3%) and, therefore, a highly imbalanced dataset. This imbalance and its implications for the model will be discussed in more detail later in the paper.

[1] https://github.com/ramondalmau/metafora

TABLE I: DESCRIPTION OF THE CATEGORICAL (TOP) AND NUMERICAL (BOTTOM) FEATURES IN THE TRAIN AND TEST SETS.

Set	Train (1.6M observations)				Test (397K observations)			
Metric / Feature	**Missing**	**Unique**	**Top**	**Top freq.**	**Missing**	**Unique**	**Top**	**Top freq.**
Airport	0.00	46		0.02	0.00	46		0.02
Wind compass	0.00	17	VRB	0.09	0.00	17	VRB	0.09
CAVOK	0.00	Boolean	False	0.64	0.00	Boolean	False	0.64
Precipitation	0.00	Boolean	False	0.91	0.00	Boolean	False	0.91
Obscuration	0.00	Boolean	False	0.96	0.00	Boolean	False	0.96
Other weather	0.00	Boolean	False	0.99	0.00	Boolean	False	0.99
Thunderstorms	0.00	Boolean	False	0.99	0.00	Boolean	False	0.99
Freezing	0.00	Boolean	False	0.99	0.00	Boolean	False	0.99
Snow	0.00	Boolean	False	0.99	0.00	Boolean	False	0.99
Cumulonimbus	0.00	Boolean	False	0.95	0.00	Boolean	False	0.95
Day of week	0.00	7	Thursday	0.14	0.00	7	Monday	0.15
Hour	0.00	19	10	0.06	0.00	19	8	0.06
Month	0.00	12	March	0.09	0.00	12	December	0.10
Metric / Feature	**Missing**	**5^{th} perc.**	**Median**	**95^{th} perc.**	**Missing**	**5^{th} perc.**	**Median**	**95^{th} perc.**
Scheduled # of arrivals	0.00	3	11	32	0.00	3	11	32
Scheduled # of departures	0.00	2	10	34	0.00	2	10	33
Wind speed (m/s)	0.00	1.00	3.60	8.20	0.00	1.00	3.60	8.20
Wind gust (m/s)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Visibility (m)	0.00	6K	10K	10K	0.00	6K	10K	10K
Ceiling (m)	0.00	244	3K	3K	0.00	244	3K	3K
Sky cover (oktas)	0.41	⅜	⅝	⅞	0.41	⅜	⅝	⅞

C. Model

Many machine learning models can be configured to handle binary classification tasks, from simple logistic regression to complex ANNs architectures. The model proposed in this study is based on ensemble methods, which produce a strong learner from a group of weak learners. Boosting is a well-known ensemble method that involves training a series of weak learners (such as rudimentary decision trees) sequentially. The training observations for the next learner in traditional adaptive boosting (AdaBoost) [17] are weighted based on how well the previous learners performed, i.e., observations that correspond to wrong predictions are assigned more weight in order to concentrate the model's attention on correcting them. Gradient boosting differs from AdaBoost in that, instead of assigning weights to observations based on performance, a new learner is trained at each iteration to fit the residual errors of the preceding learners. The entire ensemble is known as GBDTs model when decision trees are used as weak learners.

Gradient-boosted decision trees (GBDTs) can outperform ANNs in many practical applications, notably on tabular datasets where each row corresponds to one observation and each column represents a feature [18]. Furthermore, GBDTs are easier to interpret than ANNs and have very attractive properties such as the ability to handle missing data and categorical features with high cardinality (e.g., the airport). The decision to adopt the GBDTs model for addressing the specific problem in this study is underpinned by these compelling benefits. In this paper, the Microsoft's `lightGBM` implementation of GBDTs has been selected. The reader is referred to [19] for more information about the distinctive features of `lightGBM` when compared to other implementations of GBDTs like `XGBoost` [20] or `CatBoost` [21].

Fine-tuning the hyper-parameters of a GBDTs model primarily revolves around adjusting the number of decision trees, along with the maximum depth and number of leaves of each tree. To identify the optimal hyper-parameter configuration for the GBDTs model in terms of average precision, a grouping K-fold CV approach with $K = 5$ was employed. Akin to the train-test split, the train set was divided into the 5 folds while ensuring that all observations belonging to one airport during one date remained exclusively within a single fold.

In some problems, the relationship between certain predictors and the target is known in advance. For instance, it is well-known that, all else being equal, the lower the visibility, the higher the probability of regulation. Ideally, the model should autonomously learn these relationships. In practice, the noise in the dataset may lead the model to learn relationships that are not correct. For instance, if regulations were (by coincidence) never active in a given airport when thunderstorms were present, the model may interpret that thunderstorms decrease the probability of regulation. A human can easily identify this kind of misinterpretations, but a model requires guidance.

A simple and effective approach to address this issue consists of enforcing monotonic constraints, which ensure that certain predictors exhibit a monotonic relationship with the target. Two types of monotonic constraints are possible:

$$f(x_1, x_2, \ldots, x, \ldots, x_d) \leq f(x_1, x_2, \ldots, x', \ldots, x_d) \quad (9)$$

whenever $x \leq x'$ is a positive constraint; or

$$f(x_1, x_2, \ldots, x, \ldots, x_d) \geq f(x_1, x_2, \ldots, x', \ldots, x_d) \quad (10)$$

whenever $x \leq x'$ is a negative constraint.

TABLE II: BINARY CLASSIFICATION METRICS FOR THE GBDT MODEL TRAINED ON CLEAN DATA, EVALUATED ON BOTH NOISY AND CLEAN TEST SETS FOR AIRPORTS WITH OVER 1% POSITIVE OBSERVATIONS. PARENTHESES IN THE NOISY TEST SET INDICATE PERFORMANCE OF A MODEL TRAINED WITHOUT CL OR MONOTONE CONSTRAINTS. RESULTS FOR THE NOISY MODEL ON THE CLEAN TEST SET ARE OMITTED DUE TO POTENTIAL INFORMATION LEAKAGE, AS A SIMILAR MODEL WAS USED TO FILTER OUT THE NOISY OBSERVATIONS (SEE LAST PARAGRAPH OF SECTION II FOR FURTHER DETAILS).

Test set		**Clean (393K observations)**				**Noisy (397K observations)**			
Metric	**Proportion of positives**	**AP**	**ROC AUC**	**Precision**	**Recall**	**AP**	**ROC AUC**	**Precision**	**Recall**
Airport									
EDDF	0.02	0.51	0.96	0.75	0.31	0.37 (0.33)	0.90 (0.86)	0.46 (0.41)	0.25 (0.21)
EGKK	0.02	0.46	0.95	0.74	0.29	0.33 (0.26)	0.91 (0.86)	0.40 (0.36)	0.27 (0.23)
EGLL	0.06	0.68	0.96	0.84	0.39	0.46 (0.43)	0.92 (0.90)	0.52 (0.47)	0.34 (0.29)
EGSS	0.01	0.40	0.94	0.62	0.27	0.15 (0.14)	0.89 (0.88)	0.26 (0.26)	0.20 (0.19)
EHAM	0.07	0.72	0.94	0.82	0.53	0.53 (0.49)	0.89 (0.86)	0.56 (0.47)	0.48 (0.44)
EIDW	0.01	0.88	0.99	0.85	0.74	0.67 (0.59)	0.97 (0.93)	0.68 (0.62)	0.61 (0.60)
LFPG	0.01	0.69	0.99	0.74	0.55	0.45 (0.43)	0.96 (0.94)	0.55 (0.52)	0.41 (0.38)
LFPO	0.03	0.70	0.98	0.80	0.49	0.41 (0.40)	0.88 (0.87)	0.40 (0.42)	0.41 (0.40)
LPPR	0.02	0.78	0.99	0.81	0.65	0.52 (0.46)	0.97 (0.96)	0.42 (0.37)	0.62 (0.54)
LPPT	0.04	0.53	0.91	0.63	0.47	0.25 (0.21)	0.84 (0.83)	0.25 (0.24)	0.43 (0.41)
LSZH	0.04	0.80	0.98	0.88	0.63	0.57 (0.49)	0.95 (0.91)	0.55 (0.49)	0.59 (0.49)
LTFM	0.03	0.82	0.98	0.95	0.64	0.45 (0.43)	0.70 (0.68)	0.81 (0.79)	0.36 (0.35)

Please note that monotonic constraints can only be applied to numerical and boolean variables. In this experiment, positive monotonic constraints were applied to all boolean variables except for CAVOK, for which a negative constraint was enforced. For the numerical features, positive monotonic constraints were applied to all variables except for visibility and ceiling, which were assigned negative constraints.

IV. Results

This section presents the results of the experiment after training the GBDTs model on the dataset from the preceding section. Section IV-A discusses the results obtained from the noisy characterisation process of the labels in the dataset. Section IV-B showcases the model's performance on the noisy and clean test sets for airports with more than 1% of observations belonging to the positive class. Finally, Section IV-C employs feature attribution methods to interpret the model's decisions.

A. Noise characterisation

As explained in Section II, the matrix of out-of-sample predicted probabilities $\hat{\boldsymbol{P}}$ and the vector of noisy labels $\tilde{y}$ can be used to characterise the class-conditional label noise. The negative and positive per-class thresholds calculated with Eq. (5) to determine which cell of the confident joint matrix $\boldsymbol{C}$ each observation is assigned to are as follows:

$$t_0 = 0.99; \quad t_1 = 0.77,$$

which suggest that the model was more confident in true negative observations (i.e., correct prediction about the absence of regulation) than in true positive observations. The confident joint matrix, $\boldsymbol{C}$, for the train and test sets are, respectively:

$$\boldsymbol{C}_{\text{train}} = \begin{bmatrix} 1.4\text{M} & \mathbf{11K} \\ \mathbf{5.3K} & 6.2\text{K} \end{bmatrix}; \quad \boldsymbol{C}_{\text{test}} = \begin{bmatrix} 358\text{K} & \mathbf{2.9K} \\ \mathbf{1.2K} & 1.6\text{K} \end{bmatrix},$$

The analysis revealed that 5.3K out of 20.3K regulated 1-hour windows and 11K out of the 1.6M non-regulated windows in the train set were mislabelled.

To prevent the model from learning noise, these observations were pruned during training, as described in Section II. In the test set, 1.2K out of 5K regulated 1-hour windows and 2.9K out of the 395K non-regulated windows were identified as mislabeled and subsequently removed. This resulted in a clean test set used for the performance evaluation that follows.

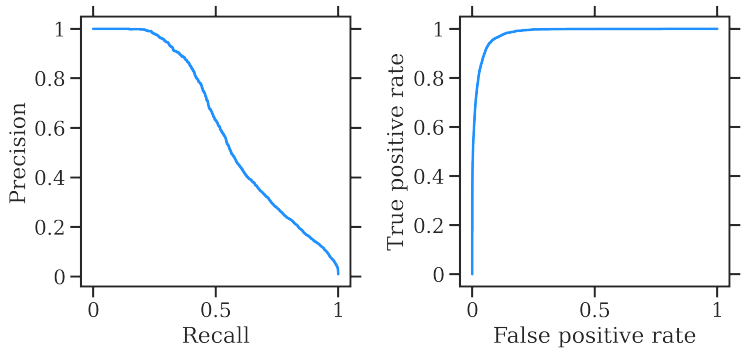
B. Performance metrics

Threshold metrics are used to summarise the fraction of times when a predicted class (positive or negative) does not match the actual class. Well-known threshold metrics are the precision and the recall. The precision answers to the question: *what proportion of positive predictions was actually correct?*, whereas the recall answers to the question: *what proportion of actual positives was predicted correctly?*.

In a binary classification task, a certain cut-off determines whether the prediction belongs to the positive or negative class based on the predicted probability. The default value for the cut-off is 0.5. This cut-off, however, is not fixed and can be fine-tuned to adjust the performance of the model: reducing its value would increase the recall while lowering the precision and vice versa, and this is exactly what rank metrics capture.

The receiver operating characteristic (ROC) and the precision-recall (PR) curves are two diagrams widely used to assess the performance of binary classifiers. Essentially, the ROC curve captures the trade-off between the true positive rate and the false positive rate for different cut-offs. Analogously, the PR curve describes the trade-off between the precision and the recall. Furthermore, each curve can also be aggregated with a unique area under the curve (AUC) score.

When dealing with imbalanced datasets, the ROC AUC score can report an overly optimistic view of performance. In this situation, the PR AUC score (also known as average precision, or AP) is often preferred because it focuses on the minority class. Table II shows the threshold (assuming a cut-off of 0.5) and rank metrics of the GBDTs model trained on the clean train set, evaluated on both noisy and clean test sets for airports with more than 1% of positive observations.



(a) Precision-recall (b) Receiver operating characteristic

Figure 1: Precision-recall and receiver operating characteristic diagrams in the clean test set.

For airports with a small number of positive observations (lower than 1%), the classification metrics are not statistically representative and have been omitted from the table for the sake of clarity in the interpretation and discussion of results.

Based on the metrics presented in Table II, the performance on the noisy test set indicates a scope for improvement, except for LTFM, which demonstrates relatively acceptable precision and recall. Notably, the performance is particularly modest for EGSS and LPPT, as both airports exhibit a precision of around 25% and a recall well below 50%. This modest performance highlights that the periods when an ATFM regulation due to weather was observed are not always correlated with the observed weather and scheduled traffic demand. Consequently, a machine learning model faces challenges in (1) learning the relationship between input features and the target value and (2) dealing with the noise of uncertain labels, leading it to learn patterns that may not be entirely correct.

Table II also demonstrates that the model's performance, when trained on the clean train set and evaluated on the clean test set, is significantly improved. Notably, the precision remains in the range of 62% to 95%, and the recall ranges from 27% to 74%. Remarkably, the removal of noisy labels from the dataset with CL yields a more pronounced improvement in precision compared to recall. Results also show that, even in the noisy test set, the model trained with CL and monotone constraints outperforms a model trained on the noisy data without monotone constraints (i.e., the classical approach).

The results suggest that by removing from the dataset the observations in which a regulation was active but maybe was not fully effective, as well as the observations in which a regulation was not active but could have been beneficial – based on evidence from the 1.6M observations used for training –, the model is capable of effectively capturing the actual relationship between weather conditions, scheduled demand, and the likelihood of regulation due to adverse weather.

Figure 1 complements Table II by illustrating the PR and ROC curves on the clean test set. As expected, the ROC curve appears very close to that of a perfect classifier, but this outcome is not surprising given the highly imbalanced dataset, leading to an overly optimistic ROC curve.

On the other hand, the PR curve provides a more equitable illustration of the model's performance. Notably, the PR curve reveals a region with precision close to 100%, where decreasing the cut-off increases the recall to approximately 25% without significantly penalising precision. For the default cut-off of 0.5, the overall precision and recall in the clean test set, considering the 393K observations from 46 airports after removing potential noise, are 77% and 44%, respectively. These metrics in the noisy test set (i.e., considering the 397K observations), are much worse: 42% and 33%, respectively.

C. Model interpretation

The preceding section suggested that the model effectively learned the relationship between weather conditions, scheduled traffic demand, and the likelihood of ATFM regulation due to weather from the observations in the clean train set. This section takes it a step further by interpreting the predictions of the model for all observations in the clean test set.

Principles from game theory can be used to interpret the prediction of a machine model for a given observation, assuming that each feature is a player of a game and the output of the model (i.e., the prediction) is the payout. Let us consider the following scenario: all players participate in the game, and they join the game in a random order. The attribution of a player is the average change in the payout received by players in the game when he or she joins them. More formally, the Shapley value $\phi_i(\boldsymbol{x}, \boldsymbol{\theta})$ of the feature i for a given input vector $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$ is defined as the expected marginal contribution of i to the prediction across all possible feature permutations [18]. In other words, the Shapley value quantifies how much each feature contributes to the model output on average when considering all possible subsets of features that include i. At present, Shapley values are very popular.

In practical applications, computing Shapley values precisely is a computationally intensive task. To address this issue, a new explanation method called `TreeExplainer` has been developed for tree-based models, such as GBDTs. The `TreeExplainer` can approximate Shapley values in polynomial time and was used in the paper referenced. Further details about the `TreeExplainer` can be found in [18].

Figure 2 shows the Shapley values distribution for the features related to weather of the GBDTs model, considering all observations in the clean test set. The y-axis indicates the name of the features, in order of mean absolute Shapley value from the top to the bottom. Each dot in the x-axis shows the Shapley value of the associated feature on the prediction for one observation, and the colour indicates the magnitude of that feature: red indicates high, while blue indicates low. For categorical features, the colour has no meaning. Note that the Shapley values are expressed in *logit*, not in terms of probability. In any case, a positive Shapley value indicates that the feature contributes to the prediction for the observation by increasing the probability relative to the expected value in the train set, while a negative value indicates the opposite.

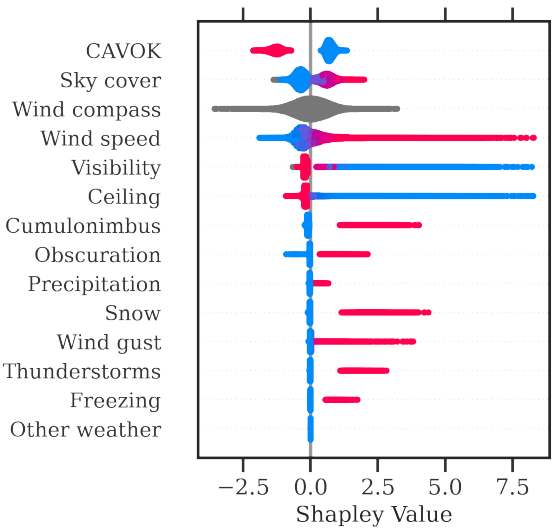


Figure 2: Distribution of Shapley values in the clean test set.

According to Fig. 2, the GBDTs model, trained on the clean train set and subject to monotone constraints, has learned patterns that are intuitive and evident from a human point of view. For instance, when the ceiling and visibility are considered *okay* (i.e., CAVOK status is true – red), the probability of ATFM regulation due to weather decreases significantly. The impact of wind speed is negative when its value is low but becomes very positive (up to +8) in the presence of strong winds. The opposite reasoning applies to visibility and ceiling. As expected, high visibility and ceiling have a negative, if not null, impact on the model's prediction.

Regarding other boolean features, such as the presence of cumulonimbus or snow, all tend to increase the probability of ATFM regulation due to weather when their status is true, albeit with very different magnitudes. For instance, the presence of precipitation has a marginal impact on the model's output, whereas snow could increase the logit up to 5 for some specific predictions. Notably, the Shapley value of a feature in an observation also depends on the values of the other features.

Figure 2 also demonstrates the effect of the monotone constraints, which guarantee a consistent attribution of features and result in a model that is robust to noise in the train set.

D. Illustrative example

This section illustrates the usability of the model with a specific example, showcasing the predictions at Zurich airport on 14th February 2023. It is important to note that this specific combination of airport and date belongs to the test set.

Figure 3 shows the predicted probability of ATFM regulation due to weather (red) and actual ATFM regulation due to weather activation status (blue) during that day. On this specific day, the morning and evening were impacted by severe obscuration caused by freezing fog. This is supported by the meteorological reports displayed in Table III.

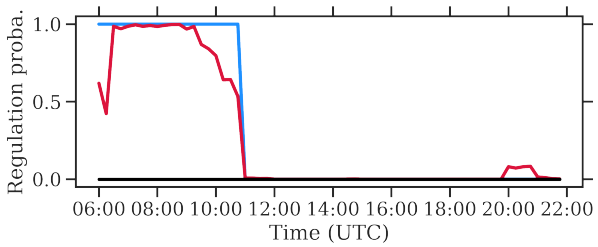


Figure 3: Predicted probability of ATFM regulation due to weather (red) and actual ATFM regulation activation status (blue) at Zurich airport on 14th February, 2023.

TABLE III: METEOROLOGICAL REPORTS AT ZURICH AIRPORT ON 14TH FEBRUARY, 2023. THE METEOROLOGICAL REPORTS WHERE THE VISIBILITY WAS LOWER THAN 800 M (I.E., CAT II/IIIA/IIIB PRECISION APPROACH CONDITIONS) ARE HIGHLIGHTED IN BOLD.

METAR LSZH 142150Z VRB02KT **0400** FZFG VV002 M01/M01 Q1032
METAR LSZH 142120Z VRB02KT **0250** FZFG VV001 M01/M01 Q1032
METAR LSZH 142050Z VRB02KT **0200** FZFG VV001 M01/M01 Q1032
METAR LSZH 142020Z 31003KT **0300** BCFG VV002 M02/M02 Q1033
METAR LSZH 141950Z 31003KT 4500 BR FEW002 M02/M02 Q1033
METAR LSZH 141920Z 29002KT 5000 BR NSC M02/M03 Q1033
METAR LSZH 141850Z VRB01KT 6000 NSC M00/M01 Q1033
METAR LSZH 141820Z 31005KT 7000 NSC M00/M01 Q1033
METAR LSZH 141750Z 32003KT 8000 NSC M00/M01 Q1033
METAR LSZH 141720Z 32004KT 9000 NSC 00/M02 Q1033
METAR LSZH 141650Z 33004KT 9000 NSC 01/M01 Q1033
METAR LSZH 141620Z 35004KT 9000 NSC 04/00 Q1033
METAR LSZH 141550Z 32004KT 9000 NSC 05/01 Q1033
METAR LSZH 141520Z 31004KT 9000 NSC 05/01 Q1033
METAR LSZH 141450Z 30004KT 270V340 9000 NSC 05/01 Q1033
METAR LSZH 141420Z 30004KT 270V330 8000 NSC 04/01 Q1033
METAR LSZH 141350Z 29005KT 260V330 8000 NSC 04/01 Q1033
METAR LSZH 141320Z 30005KT 260V320 8000 NSC 04/00 Q1034
METAR LSZH 141250Z 28005KT 250V320 8000 FEW007 03/01 Q1034
METAR LSZH 141220Z VRB03KT 6000 SCT005 SCT008 02/00 Q1034
METAR LSZH 141150Z VRB02KT 5000 BR FEW004 BKN006 02/M00 Q1035
METAR LSZH 141120Z VRB02KT 4000 BR FEW003 BKN005 01/M01 Q1035
METAR LSZH 141050Z VRB03KT 2500 BR BKN003 OVC005 01/M00 Q1036
METAR LSZH 141020Z VRB02KT 2500 BR BKN003 OVC004 00/M01 Q1036
METAR LSZH 140950Z VRB01KT 1800 PRFG OVC003 M00/M01 Q1036
METAR LSZH 140920Z VRB03KT 1200 PRFG VV003 M00/M01 Q1036
METAR LSZH 140850Z VRB01KT 0900 FZFG VV002 M00/M01 Q1036
METAR LSZH 140820Z VRB01KT **0600** FZFG VV002 M01/M01 Q1037
METAR LSZH 140750Z VRB02KT **0600** FZFG VV002 M01/M01 Q1037
METAR LSZH 140720Z VRB03KT **0600** FZFG VV002 M01/M01 Q1037
METAR LSZH 140650Z VRB02KT **0600** FZFG VV002 M01/M01 Q1037
METAR LSZH 140620Z VRB02KT **0700** FZFG VV003 M01/M01 Q1036

According to Fig. 3, the model effectively captured the ATFM regulation due to weather active from 6AM to 11AM.

It is interesting to observe how the predicted probability decreases from 9AM to 11AM as the visibility conditions slowly improve. The model also predicted a small probability of ATFM regulation due to weather from 8PM to 9PM, approximately, driven by a significant drop in visibility. Even though the weather conditions were similar, if not worse, than during the morning period, the probability of ATFM regulation due to weather was much lower due to the low traffic demand. The model learned from historical data that when the demand is low, no ATFM regulation is required even with bad weather.

V. Conclusions

This paper presents a machine learning model explicitly designed to capture the likelihood of air traffic flow management (ATFM) regulations due to weather based on weather conditions and traffic demand. To address the inherent noise in the dataset labels, arising from decisions made in advance by operators using uncertain information, confident learning techniques are proposed. The experiments demonstrate satisfactory model performance for major European airports that frequently encounter adverse weather conditions. The model effectively captures the relationship between weather conditions, traffic demand, and the likelihood of ATFM regulations.

The main objective of this model is to assist operators in determining the effectiveness of implementing regulations and to aid airlines in predicting potential delays or airborne holdings resulting from adverse weather. By enhancing the estimation of ATFM regulation likelihood, the model offers a valuable tool for operators to make more informed decisions and optimise airport operations during adverse weather conditions.

The integration of machine learning techniques, confident learning, and monotonic constraints results in a robust and reliable machine learning model capable of accurately estimating the probability of ATFM regulations due to weather. Confident learning and monotone constraints, however, are not restricted to this particular application, and the authors encourage machine learning practitioners in air traffic management (ATM) to implement these methods whenever facing noise in the data.

Acknowledgements

The authors would like to acknowledge the contribution of Sara MESON-MANCHA and Thierry DE LANGE from the EUROCONTROL's Aviation Intelligence Unit (AIU) for kindly providing the data, as well as for their priceless recommendations and ideas for this study. Special thanks also go to Rocío BARRAGÁN-MONTES from the Airports Unit, as well as all the airports and airlines involved in this project, for their unconditional support and operational expertise.

References

[1] S. Choi and Y. J. Kim, "Artificial neural network models for airport capacity prediction," *Journal of Air Transport Management*, vol. 97, p. 102146, 2021.

[2] M. Schultz, S. Reitmann, and S. Alam, "Predictive classification and understanding of weather impact on airport performance through machine learning," *Transportation Research Part C: Emerging Technologies*, vol. 131, p. 103119, 2021.

[3] S. Wang, B. Yang, R. Duan, and J. Li, "Predicting the airspace capacity of terminal area under convective weather using machine learning," *Aerospace*, vol. 10, no. 3, 2023.

[4] R. Dalmau, J. Attia, and G. Gawinowski, "Modelling the impact of adverse weather on airport peak service rate with machine learning," *Atmosphere*, vol. 14, no. 10, 2023.

[5] A. Jardines, M. Soler, and J. García-Heras, "Estimating entry counts and atfm regulations during adverse weather conditions using machine learning," *Journal of Air Transport Management*, vol. 95, p. 102109, 2021.

[6] S. Mas-Pujol, E. Salamí, and E. Pastor, "Predict atfcm weather regulations using a time-distributed recurrent neural network," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, pp. 1–8, 2021.

[7] O. Lattrez, R. Barrag'an-Montes, and M. Michalski, "Predicting airport atfm regulations using deep convolutional networks," in *12th SESAR Innovation Days (SID)*, (Budapest, Hungary), 2022.

[8] R. Patriarca, F. Simone, and G. Di Gravio, "Supporting weather forecasting performance management at aerodromes through anomaly detection and hierarchical clustering," *Expert Systems with Applications*, vol. 213, p. 119210, 2023.

[9] R. Dalmau and G. Gawinowski, "Learning with confidence the likelihood of flight diversion due to adverse weather at destination," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5615–5624, 2023.

[10] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*, (Lake Tahoe, Nevada), p. 1196–1204, 2013.

[11] B. van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Proceedings of the 28th International Conference on in Neural Information Processing Systems (NIPS)*, (Montreal, Canada), pp. 10–18, 2015.

[12] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2233–2241, 2017.

[13] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Cost-sensitive learning with noisy labels," *Journal of Machine Learning Research*, vol. 18, no. 155, pp. 1–33, 2018.

[14] Z. C. Lipton, Y. Wang, and A. J. Smola, "Detecting and correcting for label shift with black box predictors," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, vol. 80, (Stockholm, Sweden), pp. 3128–3136, 2018.

[15] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, (Stockholm, Sweden), 2018.

[16] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, p. 1373–1411, 2021.

[17] R. E. Schapire, "Explaining adaboost," in *Empirical inference*, pp. 37–52, Springer, 2013.

[18] S. M. Lundberg, G. Erion, H. Chen, *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nature machine intelligence*, vol. 2, pp. 56–67, 2020.

[19] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 3146–3154, Curran Associates, Inc., 2017.

[20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *22nd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[21] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Proceedings of the 31st Advances in Neural Information Processing Systems (NIPS)*, (Montréal, Canada), 2018.