Predicting Airport ATFM Regulations using Deep Convolutional Neural Networks

Olivier Lattrez
Data Science
Brainjar
Leuven, Belgium

Rocío Barragán Montes
Airports Unit
EUROCONTROL
Brussels, Belgium

Mateusz Michalski
Business Analysis
ATOS
Brussels, Belgium

***Abstract*—Airport ATFM regulations are a source of inefficiencies in European aviation. Providing a timely and accurate regulation prediction to the NMOC will contribute to better situational awareness. It will allow to anticipate collateral issues and help to coordinate preventive measures that can avoid ATFM regulations from being implemented. In this paper, a Deep Convolutional Neural Network that was trained to predict the probability of an ATFM regulation at the airport level is presented. The model, which showed promising results, has been put in operation in trial mode since the summer of 2022 and has been providing valuable insights during the pre-tactical and tactical phases on a daily basis.**

***Keywords*—ATFM Regulations - Airport - Convolutional neural networks - Time series - ATFM Delays**

I. Introduction

Over the years, European aviation has been growing rapidly. Between 2013 and 2019, air traffic increased, reaching 11.1 million Instrumental Flight Rule (IFR) flights in 2019 in the European Civil Aviation Conference (ECAC) area, with an average of 30.4 thousand flights per day [1]. Due to the COVID-19 outbreak, an enormous reduction of flights was observed all over Europe in 2020 and 2021 [2]. The European network is now in the process of recovering to previous levels. Moreover, forecasts predict that by 2050 the number of IFR flights will increase to 43.7 thousand flights per day, giving around 16 million flights in total per year [3].

EUROCONTROL's Network Manager Operation Centre (NMOC) is responsible for providing Air Traffic Flow and Capacity Management (ATFCM) services to the ECAC states. If the capacity drops below the ATFCM Daily Plan (ADP) or the traffic demand exceeds the available capacity, the Flow Management Position (FMP) from the Air Navigation Service Providers (ANSPs) may coordinate with the NMOC on the traffic demand tactically (on the day of operations, D) or pre-tactically (on the day before, D-1). It is done by applying an Air Traffic Flow Management (ATFM) regulation to a specific airspace or airport [4]. In the airspace the factors that intervene in the conciliation of the demand and the capacity are airside constraints such as traffic flows, flight levels or open sectors. However, in the airport the maximum available capacity is constrained by both airside and landside constraints. Moreover, the AirPort Operations Centers (APOCs), which bring together all the different actors in the airport, also play a role in the Collaborative Decision Making (CDM) process of implementing an ATFM regulation. The Airport Function (AF) is a role in the NMOC responsible for providing the airport view to the rest of NMOC positions and opening the channel with APOCs to exchange relevant operational information.

The purpose of ATFM regulations is to limit the demand over a specific portion of airspace or an aerodrome by reducing the number of flights that can access it within a certain period of time [5]. Indeed, this approach allows mitigation of potential safety risks by capping the Air Traffic Controllers (ATCos) workload in congested airspaces or Terminal Manoeuvring Areas (TMAs) or by alleviating constrained resources at airports. However, it also brings operational issues in the form of ATFM delays. Anticipating where and when a regulation may happen might provide the room to take proactive measures to avoid it and thus save minutes of ATFM delay in the network. ATFM delays have a big economic impact on the network. In 2019, more than 1.4 million flights were delayed due to ATFM regulations with 40 percent of them for more than 15 minutes [6]. In addition, they cause additional stand occupancy times in the origin airports, a lack of predictability throughout the network, and reactionary delays. It was estimated that the average cost of one minute of an ATFM delay equals around 100 € [5], [7]. In addition, it has been estimated that each impacted flight might generate an extra cost of several thousand euros. Based on Cook and Tanner's research [8], in 2010, all ATFM delays might have caused a loss of 1,250 million euros caused by an unplanned increase in the costs of fuel, maintenance, fleet, crew, passengers and reactionary delay. Their further research from 2015 [9], shows that the average cost of ATFM delay can amount to almost 2,000 € per flight. Considering the expected increase in traffic, the balance between demand and Air Traffic Control (ATC) capacity will be a challenge for all aviation partners in the years to come.

A. Machine learning techniques addressing the ATFM problem

Recently, there has been an increasing amount of research with regard to ATFM delay prediction and ATFM regulations. Various studies have achieved increasingly accurate results applying diverse machine learning algorithms. Dalmau et al. [10] proposed a Recurrent Neural Network (RNN) model to predict the evolution of delays for regulated flights. For each flight impacted by the regulation, the model can foresee the future delay progress. Their model which has an accuracy of 75% classifies whether the delay will increase, reduce or remain stable. Sanaei et al. [11] used a Deep Convolutional Neural Network (DCNN) to predict the total time of ATFM delays and the number of involved flights. The researchers concluded that the DCNN performed 50% better compared to their Random Forest (RF) baseline model, achieving a mean absolute percentage error of 22% and 14% for the delay and delayed traffic respectively. One of the key impacts on network resiliency is the effect of ATFM regulations caused by weather conditions. Based on historical air traffic and meteorological data of the Maastricht Upper Area Control Centre (MUAC), Jardines et al. [12] developed several machine learning models (Decision Tree, Linear Regression, Random Forest, and Neural Network) to predict the number of regulated entry counts and activation of the weather regulation. Mas-Pujol et al. [13] proposed a Convolutional Neural Network (CNN) and RNN models for identifying ATFM Regulations over the MUAC region. Both models achieved around 80% accuracy. Finally, in another study [14] they focused on predicting ATFM Regulations in different traffic volumes of MUAC and Reims regions. Their RNN-CNN model achieved an average accuracy of nearly 89%.

B. Background on Convolutional Neural Networks

The idea of CNNs was first introduced by Fukushima & Kunihiko in 1980 [15]. The presented Neocognitron architecture is widely regarded as the first predecessor of CNNs. The research that is further built on the work of Hubel and Wiesel [16] was originally built for unsupervised pattern recognition tasks such as handwritten character recognition. Within their research, they presented the concept of cascading simple (S-cells) and complex cells (C-cells), which later formed the basis of convolutional and downsampling layers as are known today. Both the development of LeNet-5 [17] and the Shift-Invariant Artificial Neural Network (SIANN) [18] continued working on these concepts by introducing backpropagation for several image recognition tasks. These very basic convolutional neural networks formed the inspiration for later frameworks such as AlexNet [19], VGGNet [20] and the current state-of-the-art ResNet architecture [21]. Although CNNs were initially developed within the computer vision domain, they also found wide adaptation over time within other fields such as automatic speech recognition, real-time ElectroCardioGram (ECG) monitoring, or vibration-based structural damage detection to only name a few [22]. As opposed to the two-Dimensional (2D) CNNs that are used for image recognition, one-dimensional CNNs were developed for those applications. In a one-dimensional CNN, the expected input is three-dimensional (samples, time steps, features). Contrary to 2D CNNs, the convolutional operations are only applied over the time step dimension and the convolutional layer will output a specified amount of feature maps for every iteration over the time step dimension. Compared to popular time-series modeling alternatives such as RNN models, Long-Short-Term Memory (LSTM) models, and Gated Recurrent Unit (GRU) models, CNN's deal with time-series problems both more efficiently and effectively in a wide variety of tasks [23], [24]. CNNs require fewer parameters, are less prone to the vanishing/exploding gradient issue, and are better at finding local spatial patterns. Another popular approach is to combine both LSTM and CNN layers into a hybrid LSTM-CNN model. A study conducted by Rajagukguk et al. [25] showed that their LSTM-CNN model outperformed all three RNN, LSTM, and GRU standalone models with regard to predicting solar energy. Finally, Borovykh et al. [26] adapted the convolutional autoregressive WaveNet architecture by stacking layers of increasing dilated convolutions so that it could be applied successfully to time series prediction, and act as a strong baseline for time series forecasting.

C. Outline

The studies mentioned in section A focused on the problem of ATFM Delays and ATFM Regulations from either an airspace perspective (evaluating the entire network or airspace regions) or from an aircraft perspective (modeling ATFM Delays for each flight individually). According to the best of our knowledge, there are no particular studies targeting the ATFM Regulations explicitly at the airport level. Since the models from section A were trained for very specific tasks and their input features were transformed accordingly, they can not be easily converted to target the problem at the airport level, as it would require both architectural changes and additional transformation of the input features. In this work, we present a DCNN to predict airport ATFM regulations. In section II, we discuss the available data, input features, and output targets. Section III covers the model architecture, preprocessing steps, and training process. The results and deployment are depicted in section IV while section V summarizes the conclusion and the next steps.

II. Data

The model has been trained on a subset of the 91 largest aerodromes in Europe. Originally, it had been trained on 2018-2019 data. Due to the COVID-19 pandemic, there was a huge shift in the data as traffic disappeared completely at first and then resumed unevenly. New problems arose caused by staffing issues, sanitary measures, and other additional processes at the airports. It was decided to remove the noisy data between March 2020 and May 2021 and include data starting in June 2021, when traffic resumed consistently. The last month of data considered in the training dataset for the moment is June 2022. Since it is a time series problem, the data needs to be transformed accordingly. In this work, the model was trained to predict 24 time steps (hours) ahead. To make it more digestible, most of the logic and pseudo-code in this section covers operations for a single time step. In practice, it is extended to 24 time steps.

A. *Output target*

Predicting whether or not a regulation is in effect can be described as a binary classification problem. More specifically, this work focuses on predicting the probability of a regulation at a given airport for 24 time steps in the future. In this research, some assumptions were made prior to the modeling. First of all, regulations can be classified into various categories. In this paper, only four categories were considered: weather regulations, ATC capacity regulations, aerodrome capacity regulations, and industrial action regulations. Other categories were too infrequent or unpredictable to be foreseen by a data-driven predictive model. The second assumption is that only regulations which apply to ground level and can be attributed to a single airport are considered of interest. Additionally, it was decided to discard all regulations which were canceled prior or lasted shorter than 20% of their original duration. Finally, all the TMA regulations were removed as well, since they are more related to airspace regulations and often affect more than one airport.

The first category of interest is weather regulations. They are caused by weather phenomena such as low visibility or wind conditions. The second category is the ATC capacity regulations, which originate from having to process more flights than the ATCos can safely handle. Aerodrome capacity regulations is another category. Construction works in the apron and terminal or ongoing events at a certain airport are the main cause of those regulations. The final category concerns regulations caused by industrial actions. As opposed to the previous three categories, these regulations are quite rare but nevertheless have a huge impact on airport operations. Of course, in reality, it is more ambiguous and there exists a consistent overlap between all four categories. Nevertheless, the classification of the predicted regulation category is not of interest and hence not in the scope of this work.

B. *Input features*

In order to predict the target variable, a variety of features were selected to tackle this complex problem. For each airport, all input features in this section are transformed into 24 time steps with a one-hour interval. Moreover, all features are transformed by only taking into account current or forecasted values for the day of prediction. This implies that no historical data is considered to predict whether or not a regulation will be in effect at a certain airport.

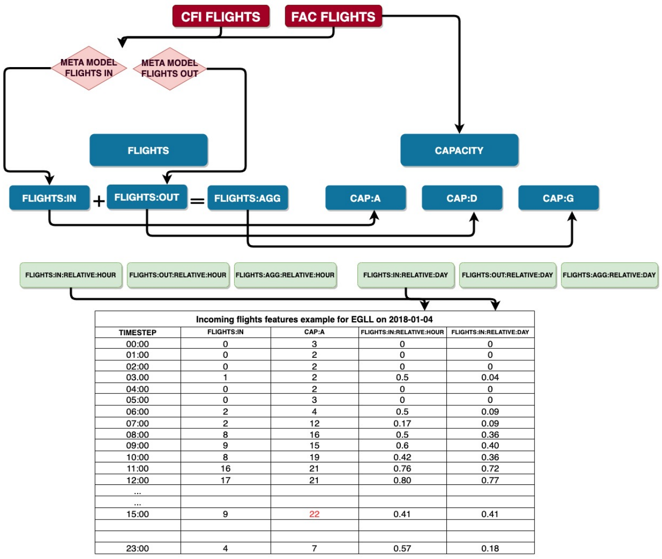
1) Flights: The first category of features that is considered to have a major impact is the number of scheduled flights. Naturally, the probability of an ATC capacity regulation at a certain airport is directly impacted by the total number of scheduled flights, especially if they are above a certain variable threshold: the airspace/airport nominal capacity. The data is transformed into a time series by taking the sum of the incoming flights (determined by either the Initial Off-Block Time or the Actual Off-Block Time) and the outgoing flights (determined by either the Actual Time of Arrival or the Estimated Time of Arrival) for each time step for each airport. To estimate the scheduled number of flights, the Consolidated Flight Intentions (CFI) were used in the first stage. Even though this is considered the best available source, it also has its limitations. For the non-coordinated airports, the airport slots are not directly included in the CFIs, but indirectly derived from airline slots which lead to a severe underestimation compared to the actual number of flights. In an effort to mediate this problem, two meta-models were trained to predict the scheduled number of incoming and outgoing flights more accurately. More specifically, the models have been trained to predict the actual number of flights based on a combination of the CFIs, the historical actual number of flights (FAC:FLIGHTS), and the cyclical features described in table I. They expect 24 time steps as input and return 24 time steps as output. To normalize the number of flights, it was decided to divide the number of flights for each time step by the maximum historical number of flights for each airport. The categorical variables were One-Hot encoded, and to represent the cyclical nature of months, the month variable was transformed by taking both the sine and the cosine of the radian transformation. A more in-depth example has been included in table I for the incoming number of flights. It follows that the same logic is applied to estimate the outgoing number of flights.

TABLE I. OUTPUT TARGET AND INPUT FEATURES OF THE INCOMING FLIGHTS' PREDICTIONS META-MODEL FOR A SINGLE TIME STEP

Feature	Description	Transformation
FAC:IN t	# Actual incoming flights at t	# Max flights
AIRPORT	ICAO Code of airport	OHE
WEEKDAY	Weekday of t	OHE
$\sin(2\pi Month/12)$	Normalized sine of month at t	
$\cos(2\pi Month/12)$	Normalized cosine of month at t	
CFI:IN t	# Predicted incoming flights at t	# Max flights
CFI:IN t-1	# Predicted incoming flights at t-1	# Max flights
CFI:IN t-2	# Predicted incoming flights at t-2	# Max flights
CFI:IN t-3	# Predicted incoming flights at t-3	# Max flights
CFI:IN t-4	# Predicted incoming flights at t-4	# Max flights
CFI:IN t-5	# Predicted incoming flights at t-5	# Max flights
CFI:IN t-6	# Predicted incoming flights at t-6	# Max flights
CFI:IN t-7	# Predicted incoming flights at t-7	# Max flights
FAC:IN t-1	# Actual incoming flights at t-1	# Max flights
FAC:IN t-2	# Actual incoming flights at t-2	# Max flights
FAC:IN t-3	# Actual incoming flights at t-3	# Max flights
FAC:IN t-4	# Actual incoming flights at t-4	# Max flights
FAC:IN t-5	# Actual incoming flights at t-5	# Max flights
FAC:IN t-6	# Actual incoming flights at t-6	# Max flights
FAC:IN t-7	# Actual incoming flights at t-7	# Max flights

To verify the accuracy of the meta-models, the mean squared error (MSE) was calculated on a holdout set. For the incoming number of flights, the MSE improved from 10.06 to 3.01. Similarly, the MSE improved from 8.04 to 2.20 for the outgoing number of flights. These results indicated significant improvements over the original CFIs. Naturally, the effect is the largest for the non-coordinated airports but also the estimations for the coordinated airports slightly improved.

2) Capacity: A second important source is the capacity of an airport. Intuitively, it is understood as the maximum number of flights that ATCos providing service to a position, an airspace sector, or an airport can handle. Unfortunately, that information is often not disclosed or accurate. Therefore, this work proposes an estimation based on historical data for 2018-2022. In this paper, maximum capacity at a given airport is calculated as the 99 percentile of the actual total flights on days when there has not been a regulation. Since it is assumed that capacity varies throughout the day, the process is repeated for each hour. This results in 24 capacity estimations for each airport instead of one static number. This logic has been applied to the incoming number of flights (featured as the maximum arrival capacity CAP:A), the outgoing number of flights (featured as the maximum departure capacity CAP:D), and the aggregated number of flights (featured as the maximum global capacity CAP:G). To not lose the general view of the day, the number of flights is also normalized over the maximum capacity of an entire day, which is just the maximum value of the 24-hourly estimations. These capacity estimations are then merged with the output of the meta-models described in the flights' section. The meta-models first output an improved estimation of the number of scheduled incoming and outgoing flights for each airport, and those estimations are then normalized over the capacity estimations for each airport obtained from the procedures discussed in this section. Figure 1 visualizes the entire process to produce the final six flight features that serve as input to the final model. Finally, the table shows an extensive example of the normalization technique for the incoming number of flights. Naturally, it follows that the same logic is applied to obtain the features for the outgoing and global number of flights.



Incoming flights features example for EGLL on 2018-01-04

TIMESTEP	FLIGHTS:IN	CAP:A	FLIGHTS:IN:RELATIVE:HOUR	FLIGHTS:IN:RELATIVE:DAY
00:00	0	3	0	0
01:00	0	2	0	0
02:00	0	2	0	0
03.00	1	2	0.5	0.04
04:00	0	2	0	0
05:00	0	3	0	0
06:00	2	4	0.5	0.09
07:00	2	12	0.17	0.09
08:00	8	16	0.5	0.36
09:00	9	15	0.6	0.40
10:00	8	19	0.42	0.36
11:00	16	21	0.76	0.72
12:00	17	21	0.80	0.77
...				
...				
15:00	9	22	0.41	0.41
23:00	4	7	0.57	0.18

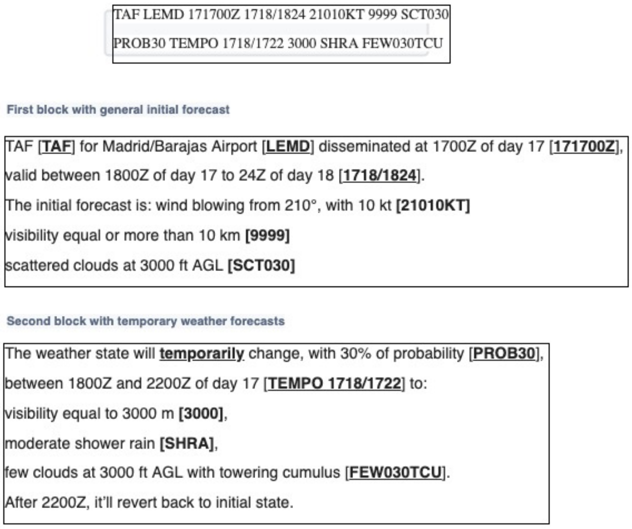
Figure 1. Illustrative example of how flight features are created

3) Airport events: Closely related to the capacity from the previous section are the most relevant events taking place at a given airport. A majority of the events can be associated with construction works, but also industrial actions and major sports events, for example, can have a big impact on the demand or the capacity of an airport. This information (filled in by airports) was extracted from the Airport Corner managed by EUROCONTROL. Airport Corner is an airport-focused web application where airports share relevant information with all stakeholders, the Network Manager, and Airlines. It contains a repository of the historically reported events with an impact on capacity or demand. Since it is not mandatory and the information often follows a free text format, the information is unfortunately often incomplete or even returned empty. Nevertheless, it still provides valuable information since the event's impact is analyzed and reported by the airport itself.

Generally, a registered event has a specified schedule for which it will have an impact on the capacity of an airport. Only events that were specified to have an impact are treated. From these events, both the category of the event and the anticipated impact (expressed in either a percentage reduction or as a maximum number of flights) are considered. If the event information contains a specified impact, the capacity estimations from the previous section are replaced with the capacity specified in the event by the airport. As a result, three additional boolean features are created. They indicate whether there is an impact on capacity at a certain time step for the arrival flow, the departure flow, or the global number of movements respectively. Any changes to the capacity are first applied before the normalization process described in the previous section.

4) Weather: Another major factor of impact is the weather. Depending on the meteorological conditions, the number of flights that can be safely managed by the ATC system varies. As a proxy for the forecasted weather, the Terminal Aerodrome Forecast (TAF) messages are being used. That data is managed and issued by The Satellite Distribution System (SADIS). A TAF message generally includes a forecast of the weather at an airport and/or its vicinity for the upcoming 24/30 hours [27]. The messages are generally rolled out every 6 hours, and can also be amended or corrected during that time span. A TAF message consists of maximum 2 blocks, for which the first is mandatory and the second one is optional. The first part of a message is a general forecast applicable to the entire interval. The second block of a message can indicate multiple specific intervals for which the weather could be different from the general forecasted weather. As the second part is more uncertain, probabilities are assigned to those forecasts. Figure 2 breaks down an example of how a TAF message is constructed.

During the preprocessing, the TAF messages are first sorted for each airport by their time of issuance. They are processed in a rolling-forward manner from the following hour to their issuance. Within this research, the TAF messages were parsed with pytaf [28]. This creates a dictionary with all parsed values and their corresponding probabilities. In the next step, a distinction is made between the general message and the optional temporary messages. The temporary part is then merged with the general part by the most logical aggregation method. While the minimum and maximum are self-explanatory, taking the set is equal to creating a list of unique elements of the merged values. For the wind direction variables finally, the temporary wind direction value was used if specified in the TAF message. Table II highlights all TAF features with their



TAF LEMD 171700Z 1718/1824 21010KT 9999 SCT030
PROB30 TEMPO 1718/1722 3000 SHRA FEW030TCU

First block with general initial forecast

TAF [**TAF**] for Madrid/Barajas Airport [**LEMD**] disseminated at 1700Z of day 17 [**171700Z**],
valid between 1800Z of day 17 to 24Z of day 18 [**1718/1824**].
The initial forecast is: wind blowing from 210°, with 10 kt **[21010KT]**
visibility equal or more than 10 km **[9999]**
scattered clouds at 3000 ft AGL **[SCT030]**

Second block with temporary weather forecasts

The weather state will **temporarily** change, with 30% of probability [**PROB30**],
between 1800Z and 2200Z of day 17 [**TEMPO 1718/1722**] to:
visibility equal to 3000 m **[3000]**,
moderate shower rain **[SHRA]**,
few clouds at 3000 ft AGL with towering cumulus [**FEW030TCU**].
After 2200Z, it'll revert back to initial state.

Figure 2. Illustrative example of a TAF message [27]

corresponding aggregation method. Next to these features, this paper also includes the corresponding probabilities for each of the weather phenomena. If the probability is not specified, it is assumed to be occurring at 100%. To reduce overfitting, the probabilities are binned into 5 categories instead of using the actual values.

TABLE II. OVERVIEW OF TAF FEATURES

Feature	**Description**	**Agg**
CLOUDS LAYER	Cloud coverage	Set
CLOUDS TYPE	Type of clouds	Set
CLOUDS CEILING	Clouds ceiling expressed in feet	Min
VERTICAL VISIBILITY	Vertical visibility expressed in feet	Min
VISIBILITY RANGE	Visibility range expressed in feet	Min
WIND:DIR:X	X component of wind angle (cos)	Temp
WIND:DIR:Y	Y component of wind angle (sin)	Temp
WIND:DIR:VRB	Variable wind direction	Max
WIND SPEED	Wind speed expressed in knots	Max
WIND GUST	Wind gust expressed in knots	Max
WEATHER INTENSITY	Weather intensity	Set
WEATHER MODIFIER	Weather modifier	Set
WEATHER PHENOMENON	Weather phenomenon	Set

5) NOTAM messages: To gather more information about ongoing operational procedures, Notice To Air Missions (NOTAM) messages are being used. According to the Federal Aviation Administration (FAA) [29], ”a NOTAM is a notice containing information essential to personnel concerned with flight operations but not known far enough in advance to be publicized by other means. It states the abnormal status of a component of the National Airspace System (NAS) – not the normal status”. The messages can contain valuable information about many different aspects such as, for example, a runway that is closed, equipment failure, degrading runway/taxiway surface conditions but also more harmless information about, for example, the grass that will be cut. Since NOTAMs always follow a fixed structure, they are relatively easy to parse. In this paper, the PyNotam [30] package was used to accomplish that. Table III shows which fields were of interest for this problem. The Q code contains 5 letters, for which the first one always is a Q. The second and third letter point to the subject of the NOTAM, while the fourth and fifth letters specify the status of the operation. The schedule field is an optional field that follows a certain grammar language, for which a custom parser was written. It refines the initial time interval for which a NOTAM is valid. If it is provided, the schedule is parsed and overwrites the initial interval.

TABLE III. OVERVIEW OF INITIAL NOTAM FIELDS

Field	**Business logic**
Aerodrome	Airport (ICAO)
ID	Unique NOTAM ID
Reference ID	Refers to the unique NOTAM ID
Type	Type of the NOTAM
Q code	NOTAM Q code
Received time	Indicates when a NOTAM was initially received
Active from	Indicates when a NOTAM becomes active
Active to	Indicates when a NOTAM is no longer valid
Schedule	Specifies periods of time for which a NOTAM is active

There exists 3 types of NOTAMs: New NOTAMs (NOTAMN), Replace NOTAMs (NOTAMR) and Cancelled NOTAMs (NOTAMC). During the preprocessing, each NOTAM is first grouped and sorted. In case of any replacement or canceled NOTAMs, the previous NOTAM expires and is either replaced as of the start time or canceled as of that time. After the initial preprocessing, a NOTAM matrix based on the Q-code is created for each airport. The matrix of dimension *time steps*number of different Q codes* is initialized with zeros. By looping over all NOTAMs, the matrix is filled with ones if a NOTAM with a Q code is active at a time step.

6) Miscellaneous: Finally, there are two more isolated features that were included. First of all, the ICAO code of an aerodrome is passed. This was done because it is assumed that airports have different critical values. These differences can be based on, for example, the size of the usual aircraft arriving, the procedures in place, or the size of the terminal, for instance. In line with it, the number of runways is also passed as a feature to the model. This was done to provide the model with a sense of the magnitude of each airport.

III. MODEL

After the data has been processed and transformed, it can be passed to the eventual model. In this section, the model architecture and the preprocessing methods are first discussed. The final part of this section elaborates on the training process.

A. Preprocessing

Since machine learning models expect only numerical input, it is important to first transform all non-numerical variables. Secondly, it is also good practice to normalize all numerical variables between the same boundaries. This has several advantages, of which quicker convergence is considered the most important one. Table IV shows how the data is normalized and encoded for the different categories.

TABLE IV. BREAKDOWN OF PREPROCESSING METHODS BASED ON VARIABLE CATEGORY

5 **Category**	**Type**	**Preprocessing method**
Regulation in effect	Numeric[Bool]	/
Flights	Numeric[Int]	MinMaxScaler
Airport events	Categorical[List]	MultiLabelBinarizer
Airport events	Numeric[Bool]	/
Weather	Numeric[Float]	MinMaxScaler
Weather	Categorical[String]	OneHotEncoder
Weather	Categorical[List]	MultiLabelBinarizer
NOTAMs	Numeric[Bool]	/
Runways	Numeric[Int]	MinMaxScaler
Airport	Categorical[String]	OneHotEncoder

B. Architecture

During this research, a variety of architectures were tried out. First, a basic LSTM with 128 units was tried. This served as an initial baseline. Next, a WaveNet architecture was examined. Although this already showed significant improvements over the LSTM architecture, its autoregressive nature constrained the learning possibilities for this specific problem too much. The model specifically had trouble with learning longer-range patterns. Due to these limitations, it was opted to switch to a hybrid LSTM-CNN model next. As this model was not autoregressive anymore, a lot of dropout (up to 80%) had to be added to avoid it from overfitting. With the addition of these regularization techniques, both the training loss and validation loss dropped significantly again. There were two main drawbacks to the model, however. First, the hybrid LSTM-CNN model was quite expensive to train due to its large amount of parameters. Secondly, one of the final layers was a concatenation layer that merged all information from previous layers, losing its sense of time at that point. Although each discussed model outperformed the previous one, a DCNN was found to be the best architecture for this problem, overcoming all of the previously discussed drawbacks. Figure 3 visualizes the final model architecture from this paper. For the input layers, the NOTAM features are separated from the other input features. This was decided because the NOTAM features were found to be prone to overfitting. Therefore, a dropout layer of 30% is first applied before all features are concatenated. The main building block of the model is the convolutional block which is repeated 7 times. This block consists of a one-dimensional convolutional layer with 64 filters, a kernel size of 3, a stride of 1, minor L2 regularization, and is padded with zeros to preserve the dimensionality of the 24 time steps. The convolutional layer is followed by a Batchnormalization layer, a dropout layer of 10%, and finally a ReLU activation layer. As with other artificial neural networks, the model tends to first learn more general features within the first layers (or blocks) and gradually learns more specific patterns as it moves deeper through the network. Right before the output layer, a final one-dimensional convolutional layer with kernel size 1 and a single filter is added, resulting in a final dimension of 24x1. Since the initial goal of this research was to predict whether or not there is going to be a regulation at a given airport on a certain day, it was also decided to output a single value representing the probability of a regulation for the entire day. As this should be a function of the hourly predictions, this was done by applying a one-dimensional GlobalMaxPooling layer over the hourly predictions.

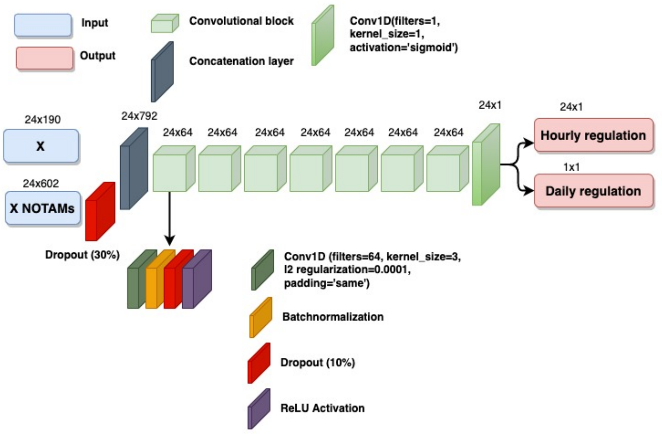


Figure 3. Model architecture

C. Training process

To evaluate the model independently, a holdout set of 5000 instances (200 dates for a selection of 25 airports) was first created. After removing those instances from the dataset, the model was trained in a 5-fold cross-fold validation. This has the advantage that all available data can be used during training while still preserving a strong estimation of the overall performance of the model. With this procedure, five different models are created, each of which may have learned different patterns. To benefit from this, these five models are stacked and the average of the predictions is the final estimation.

During training, the models were optimized for two different loss functions simultaneously. Since the target variable is binary, the Binary Cross Entropy (BCE) loss function (Eq. 1) is a very common choice. Additionally, it was opted to also include the Dice Loss (Eq. 2), which is primarily very popular in the computer vision domain. In both equations, y is defined as the ground-truth value and $\hat{y}$ as the predicted value. As opposed to the BCE, the choice for the Dice Loss is perhaps less evident. The loss was originally designed to calculate the similarity between images and is a very popular choice for class-imbalanced datasets. Given that some airports in the dataset have very few regulations, and some airports have many, it was decided to use the Dice Loss as an additional loss function to tackle this imbalance. Additionally, minor label smoothing of 0.01 was applied to both loss functions. Finally, the loss functions were slightly modified by adding an exponential increase once a prediction was above 0.8. This was done with the hypothesis that the model would then focus less on optimizing the final few percentages, but rather focus on examples that are harder to learn.

As the Dice Loss is essentially optimizing for a metric, it will heavily try to push the predictions towards 1. The BCE

Loss on the other hand generally has smoother gradients and will output more evenly distributed values. Translating this to the output variables, the Dice Loss is an excellent choice for the daily probability predictions, while the BCE makes more sense for the hourly predictions, as small offsets in the hourly predictions are not really troublesome and sometimes even welcome. The actual loss function is a concatenation of the both adapted L$_{BCE}{}^*$ and the L$_{Dice}{}^*$ (Eq. 3), for which slightly more weight was given to the hourly predictions.

$$L_{BCE}(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)) \tag{1}$$

$$L_{Dice}(y,\hat{y}) = 1 - \frac{2\sum_{i=1}^{N} y_i \cdot \hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i} \tag{2}$$

$$L_{concat}(y,\hat{y}) = 0.6L_{BCE^*}(y,\hat{y}) + 0.4L_{Dice^*}(y_{max},\hat{y}_{max}) \tag{3}$$

The loss functions have been optimized using Adam with a learning rate of 0.0007 and a decay of 0.0003. The models have been trained with a batch size of 1024 for 100 epochs with early stopping of 20 epochs. To mediate the class imbalance, the positive samples were oversampled during training. With this approach, new synthetic samples were created by slightly modifying the original data. More specifically, a randomly sampled percentage (ranging from 0 to 10%) of the input values was swapped with its nearest neighbors (on an airport level) based on the concept of Dynamic Time Warping (DTW). DTW was first introduced by Sakoe & Chiba in 1978 [31]. It has become a prominent similarity algorithm between two temporal sequences as it considers similarity irrespective of the absolute time. Prior to training, the DTW matrix between all positive instances was calculated for each airport individually. During training, the DTW matrix was searched to select one of its randomly sampled nearest neighbors. As the implementation generates new synthetic positive samples every epoch, this helped to overcome the imbalanced nature of the data due to the scarcity of regulations and eliminated the need to introduce class or sample weights.

IV. Results

In this section, results are first evaluated on the holdout set of 5000 instances, consisting of 200 days for 25 airports. The second part of this section shows an illustrative example of the output and focuses on the explainability of the models.

A. *Performance metrics*

To evaluate the performance of the model, the metrics are compared against two baselines. The first baseline is the random guess baseline, for which the predictions are randomly sampled from a uniform distribution between 0 and 1. The second baseline is the naive persistence baseline, which is the equivalent of the majority class baseline in the case of a non-time series classification. This baseline uses the value at the previous time step (t-1) as the prediction for the next time step (t). Transforming this to a daily prediction, this algorithm takes the daily prediction from the day before (t-1) as the prediction for the day itself (t). This baseline is usually very competitive, as the values from subsequent time steps tend to be closely related to each other. Translating this to regulations, it is quite likely that if there was a regulation the previous day due to bad weather, an overload of traffic, or construction works, this will still apply for the next day as well. For the hourly predictions, the hourly values from the day before are used. Table V compares the model performance (evaluated as the average prediction of the 5 models) against the baseline models. The metrics from the table show the macro average across all 25 airports considering that an airport is expected to have a regulation if the probability of occurrence is above 0.5. The binary accuracy is simply the accuracy of the positive and negative labels combined. A precision of 0.76 implies that of all the predictions regulations, 76% of them were correctly predicted. The recall of 0.76 means that all of the existing regulations, 76% were correctly recognized as a regulation. The F1-score finally is the harmonic mean of the precision and the recall.

TABLE V. Comparison of the model performance against baseline models

Model	Type	Bin. Acc.	Precision	Recall	F1
Random guess model	Daily	0.51	0.24	0.52	0.33
Naive persistence model	Daily	0.82	0.61	0.60	0.61
Stacked Conv1D model	Daily	**0.89**	**0.76**	**0.76**	**0.76**
Random guess model	Hourly	0.50	0.04	0.50	0.08
Naive persistence model	Hourly	0.95	0.33	0.33	0.33
Stacked Conv1D model	Hourly	**0.96**	**0.48**	**0.49**	**0.49**

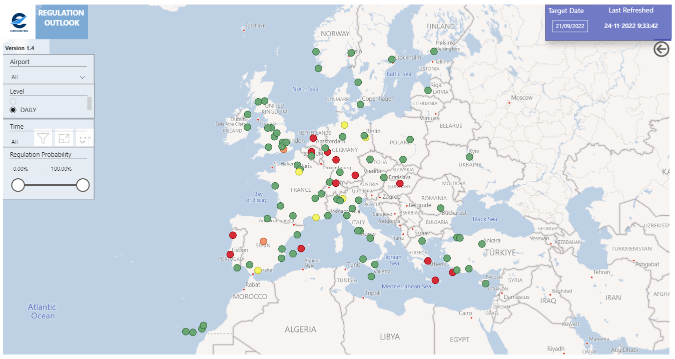
Evaluating the results shows that the model is confidently beating both baseline models. It is important to note that it does so by not considering the previous day's state as an input feature. As opposed to the naive persistence model, the model just evaluates the forecasted values using the inputs for the day itself. Compared to the most competitive naive persistence baseline, the results for the daily predictions improved for all metrics. The binary accuracy increased from 0.82 to 0.89, the precision increased from 0.61 to 0.76, the recall increased from 0.60 to 0.76, and the F1-score improved from 0.61 to 0.76. The enhancements for the hourly predictions are very similar, for which the F1-score, for example, improved from 0.33 to 0.49. Considering that we are primarily interested in a rough estimation of when a regulation will be issued, a small offset in the hourly predictions is negligible (e.g. predicting a regulation from 08:00 until 10:00 for a regulation that took place from 09:00 until 11:00 is still considered a strong prediction). Therefore, the accuracy of the hourly predictions is of less importance compared to the daily predictions.

B. Explainability

The work from this paper has been implemented as a trial version in the NMOC through the Airport Function Dashboard (AFD). As artificial neural networks are often considered black boxes, explainable AI has become increasingly important. Within the dashboard, there is the possibility to break down the feature importance for each airport individually. The ability to explain the decision process of neural networks not only makes the models more interpretable but also builds trust and increases the adaptation of machine learning models. Although many frameworks have been developed in recent years, a custom implementation had to be developed due to the model being a multi-output time series model. This implementation is based on the SHapley Additive exPlanations (SHAP) values [32], which is a concept originally from cooperative game theory. It was initially designed to assign payouts to players depending on their contribution to the total payout. This logic was later extended to the machine learning domain, where the features represent the players and the total payout is the overall prediction value. The marginal contribution of each feature is calculated by predicting once including the feature and once excluding (masking) the feature for various subsets of features. The average difference between the prediction including the feature and excluding the feature is considered the marginal contribution for that feature.

C. Deployment

The predictions in the AFD are currently updated every 3 hours with the latest information available. The main view (Figure 4) shows an interactive map of Europe with aerodromes as dots representing the daily probability of a regulation for a given target date. The various colors highlight different categories of probabilities, ranging from green (<10%), yellow (>=10% and <50%), orange (>=50% and <80%) to red (>=80%). Translating the model performance on the holdout set, on average 83% of the airports highlighted in red will be correctly labeled as a regulation. Similarly, 86% of all existing regulations will be identified among all the yellow, orange, and red-marked airports. Hovering over the dots shows the exact daily probability expressed in percentages. Double-clicking on a dot forwards the user to an aerodrome-specific page, visualized in figure 5. This page breaks down the hourly predictions for the target date and also shows the feature importance based on the SHAP values. Both the average feature importance (over the entire day) and the feature importance over time are visualized along with the hourly predictions. Figure 5 is an illustrative example of Brussels National (EBBR) with target date September 21 2022. Based on the information from the visualization, there is a very high chance of a regulation in the early morning. According to the hourly feature importance extraction for the most likely regulation at 06:00, the biggest contributor is related to visibility issues (smoke), but also the amount of traffic (emphasized by the flight features), the wind direction, and taxiway maintenance were recognized as other potential problematic factors. Comparing it against the ground truth, figure 6 shows that there was indeed a regulation activated from 05:00 until 08:00. Evaluating the cause of the regulation, low visibility ('LOW VIS') was identified as the predominant factor. As opposed to only being provided a single probability value, the NMOC users can start filtering out known issues, identifying potential regulations, and agreeing with airports on the most appropriate and efficient measures to be taken.



Target Date 21/09/2022
Last Refreshed 24-11-2022 9:33:42

Airport: All
Level: ● DAILY
Time: All
Regulation Probability: 0.00% 100.00%

Figure 4. AFD main view

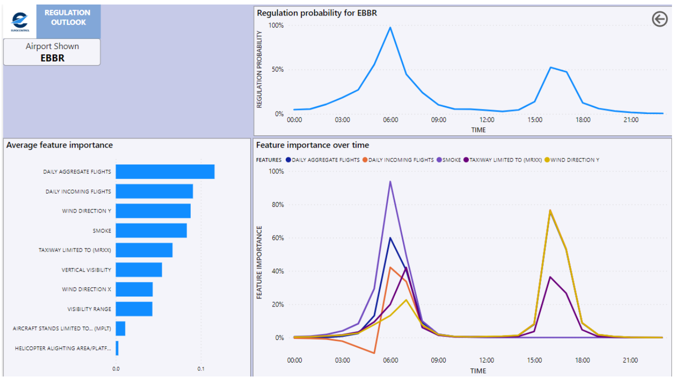
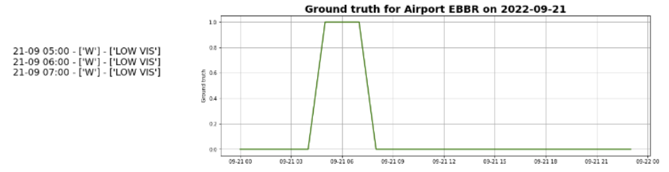


Figure 5. Illustrative Example for EBBR



```
21-09 05:00 - ['W'] - ['LOW VIS']
21-09 06:00 - ['W'] - ['LOW VIS']
21-09 07:00 - ['W'] - ['LOW VIS']
```

Figure 6. Ground truth values for illustrative example for EBBR

V. Conclusion

In this paper, a Convolutional Neural Network designed to predict the probability of an airport ATFM regulation occurrence during the pre-tactical and tactical phases is presented. To the best of our knowledge, this is the first research that focuses on predicting ATFM regulations from an airport perspective. Evaluating the model on the holdout test set shows promising performance assessment, while SHAP values for each regulation appearance support the decision with the adequate explainability. The model has been integrated into the EUROCONTROL AFD since the summer of 2022 in trial mode. Improved airport situational awareness in the NMOC increases operational performance and brings an overall mitigation of ATFM delay economic impact. Airports and airlines that are aware of potential network bottlenecks could better prepare themselves and anticipate adequate actions. Accordingly, sharing the Regulation Outlook tool with the rest of aviation stakeholders is envisaged in the future. To improve the model, future research will consider longer-term predictions (D-6), provided that accurate forecasting sources for weather and traffic are available. In addition, a categorization of the predicted regulations in terms of severity (expected minutes of ATFM delay) or causes will be explored.

Acknowledgment

This work has been funded by SESAR Wave 2, in the scope of the project PJ.04 Total Airport Management (TAM) addressing solution 28.3: Connected large airports. The collaboration of the authors Olivier Lattrez (Brainjar) and Mateusz Michalski (ATOS) with EUROCONTROL is in the framework of CRONOS Consortium - 15-140654-C.

References

[1] EUROCONTROL2019a. Performance review report: An assessment of air traffic management in europe during the calendar year 2019. [Online]. Available: https://www.eurocontrol.int/publication/performance-review-report-prr-2019

[2] EUROCONTROL2022a. Eurocontrol comprehensive assessment for wednesday, 18 august 2022. [Online]. Available: https://www.eurocontrol.int/publication/eurocontrol-comprehensive-aviation-assessment

[3] EUROCONTROL2022b. Eurocontrol aviation outlook 2050. [Online]. Available: https://www.eurocontrol.int/publication/eurocontrol-aviation-outlook-2050

[4] EUROCONTROL2022c. Atfcm operations manual. [Online]. Available: https://www.eurocontrol.int/publication/atfcm-operations-manual

[5] EUROCONTROL2019b. Atfm regulation: a power for good understanding how it works. [Online]. Available: https://www.eurocontrol.int/publication/air-traffic-flow-management-atfm-regulations-power-good

[6] EUROCONTROL2019c. Network operations report 2019. [Online]. Available: https://www.eurocontrol.int/publication/annual-network-operations-report-2019

[7] EUROCONTROL2019d. Air traffic management cost-effectiveness (ace) benchmarking report for 2019. [Online]. Available: https://www.eurocontrol.int/publication/air-traffic-management-cost-effectiveness-ace-benchmarking-report-2019

[8] A. J. Cook and G. Tanner, "European airline delay cost reference values," 2011.

[9] A. Cook and G. Tanner, "European airline delay cost reference values. 2015," *Eurocontrol: Brussels, Belgium*, 2015.

[10] R. Dalmau, B. Genestier, C. Anoraud, P. Choroba, and D. Smith, "A machine learning approach to predict the evolution of air traffic flow management delay," in *14th ATM Res. Develop. Seminar*, 2021, p. 8.

[11] R. Sanaei, B. A. Pinto, and V. Gollnick, "Toward atm resiliency: A deep cnn to predict number of delayed flights and atfm delay," *Aerospace*, vol. 8, no. 2, p. 28, 2021.

[12] 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.

[13] S. Mas Pujol, E. Salamí San Juan, and E. Pastor Llorens, "A novel methodology to predict regulations using deep learning," in *10th SESAR Innovation Days: 7th of December-10th of December, 2020, virtual event*. Single European Sky ATM Research (SESAR), 2020, pp. 1–8.

[14] S. Mas-Pujol, E. Salamí, and E. Pastor, "Rnn-cnn hybrid model to predict c-atc capacity regulations for en-route traffic," *Aerospace*, vol. 9, no. 2, p. 93, 2022.

[15] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980. [Online]. Available: https://doi.org/10.1007/BF00344251

[16] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[18] W. Zhang, J. Tanida, K. Itoh, and Y. Ichioka, "Shift-invariant pattern recognition neural network and its optical architecture," in *Proceedings of annual conference of the Japan Society of Applied Physics*, 1988, pp. 2147–2151.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021.

[23] A. P. Wibawa, A. B. P. Utama, H. Elmunsyah, U. Pujianto, F. A. Dwiyanto, and L. Hernandez, "Time-series analysis with smoothed convolutional neural network," *Journal of big Data*, vol. 9, no. 1, pp. 1–18, 2022.

[24] M. Markova, "Convolutional neural networks for forex time series forecasting," in *AIP Conference Proceedings*, vol. 2459, no. 1. AIP Publishing LLC, 2022, p. 030024.

[25] R. A. Rajagukguk, R. A. Ramadhan, and H.-J. Lee, "A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power," *Energies*, vol. 13, no. 24, p. 6623, 2020.

[26] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, 2017.

[27] I. V. A. organisation. Taf explanation. [Online]. Available: https://mediawiki.ivao.aero/index.php?title=TAF_explanation

[28] Pytaf library. [Online]. Available: https://github.com/dmbaturin/pytaf

[29] F. A. Administration. What is a notam? [Online]. Available: https://www.faa.gov/about/initiatives/notam/what_is_a_notam

[30] Pynotam library. [Online]. Available: https://github.com/slavak/PyNotam

[31] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[32] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.