# Deep Learning techniques for Demand-Capacity Balancing


SERGI MAS-PUJOL

*Telecommunications Engineer with mention in Computer Science*
*Master degree in Artificial Intelligence*

**Advisors**
DRA. ESTHER SALAMÍ SAN JUAN
DR. ENRIC PASTOR LLORENÇ



Doctorate program in Aerospace Science and Technology
*Technical School of Telecommunications and Aerospace of Castelldefels*
**Technical University of Catalonia - BarcelonaTech**

Programa de doctorat en Ciència i Tecnologia Aeroespacial
*Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels (EETAC)*
**Universitat Politècnica de Catalunya (UPC) - BarcelonaTech**

*A dissertation submitted for the degree of*
*Doctor of Philosophy*
March 2023

**Deep Learning techniques for Demand-Capacity Balancing**

**Author**
Sergi Mas-Pujol

**Advisors**
Dra. Esther Salamí San Juan
Dr. Enric Pastor Llorenç

**Reviewers**
Dr. Miquel Àngel Piera Eroles
Dr. Ramon Dalmau Codina

**Thesis committee**
Dr. Miquel Àngel Piera Eroles
Dr. Ramon Dalmau Codina
Dr. Xavier Prats Menéndez

**Doctorate program in Aerospace Science and Technology**
**Technical University of Catalonia - BarcelonaTech**
March 2023

This dissertation is available on-line at the *Theses and Dissertations On-line* (TDX) repository, which is managed by the Consortium of University Libraries of Catalonia (CBUC) and the Consortium of the Scientific and Academic Service of Catalonia (CESCA), and sponsored by the Generalitat (government) of Catalonia. The TDX repository is a member of the Networked Digital Library of Theses and Dissertations (NDLTD), which is an international organisation dedicated to promoting the adoption, creation, use, dissemination and preservation of electronic analogues to the traditional paper-based theses and dissertations **http://www.tdx.cat**

This is an electronic version of the original document and has been re-edited in order to fit an A4 paper.

**PhD. Thesis made in:**
Technical School of Telecommunications and Aerospace of Castelldefels
Esteve Terradas, 5
08860 Castelldefels (Barcelona), Spain

*Als meus pares i la meva parella*
*Silvia, Josep i Iciar*

# Contents

# List of Figures

# List of Tables

## Tables in Appendices

# List of Publications

The list of publications resulting from this PhD. work is given in inverse chronological order as follows:

### Journal Papers

- MAS-PUJOL, SERGI; DELGADO, LUIS. 2022. Prediction of ATFM impact for individual flights: A machine learning approach. *Submitted to Journal*
- MAS-PUJOL, SERGI; SALAMÍ, ESTHER; PASTOR, ENRIC. 2022. Image-Based Multi-Agent Reinforcement Learning for Demand–Capacity Balancing. *Journal of Aerospace, **9**, 599*
- MAS-PUJOL, SERGI; SALAMÍ, ESTHER; PASTOR, ENRIC. 2022. RNN-CNN Hybrid Model to Predict C-ATC CAPACITY Regulations for En-Route Traffic. *Journal of Aerospace, **9**, 93*.

### Conference Proceedings

- DELGADO, LUIS; MAS-PUJOL, SERGI; SKOROBOGATOV, GEORGY, ARGERICH, CLARA; GREGORI, ERNESTO. 2022 (Oct.). Dispatcher3 – Machine learning for efficient flight planning - Approach and challenges for data-driven prototypes in air transport. *In: Towards Sustainable Aviation Summit (TSAS)*. Toulouse (France).
- MAS-PUJOL, SERGI; DE FALCO, PAOLINO; SALAMÍ, ESTHER; DELGADO, LUIS. 2022 (Oct.). Pre-Tactical Prediction of ATFM Delay for Individual Flights. *In: 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*. Portsmouth, Virginia (USA).
- MAS-PUJOL, SERGI; DELGADO, LUIS; DE FALCO, PAOLINO. 2022 (May). Pre-tactical advice using machine learning for Air Traffic Flow Management delay estimation. *In: Airline Group of the International Federation of Operational Research Society (AGIFORs)*. Virtual event: AGIFORS.
- MAS-PUJOL, SERGI; SALAMÍ, ESTHER; PASTOR, ENRIC. 2021 (Oct.). Predict ATFCM weather regulations using a time-distributed recurrent neural network. *In: 2021 IEEE/AIAA 40th Digital Avionics Systems Conference*. Hybrid event: San Antonio, Texas (USA).

- MAS-PUJOL, SERGI; SALAMÍ, ESTHER; PASTOR, ENRIC. 2020 (Dec.). A novel methodology to predict regulations using Deep Learning. *In: 7th SESAR Innovation Days (SIDS)*. Virtual event: SESAR JU.

### Posters

- DELGADO, LUIS; MAS-PUJOL, SERGI; SKOROBOGATOV, GEORGY, ARGERICH, CLARA; GREGORI, ERNESTO. 2022 (Oct.). Dispatcher3 – Machine learning to support flight planning processes. *In: 12th SESAR Innovation Days (SIDS)*. Budapest (Hungary).

# Agraïments

Els inicis d'aquesta tesi es remunta a final del 2019, principis del 2020, durant la finalització de màster que vaig fer a la Universitat Pompeu Fabra. Era moment de decidir quina seria la nova aventura i començar a definir la meva carrera professional. En aquell moment hi havia un parell d'opcions sobre la taula: continuar treballant en el grup de recerca de la UPF o canviar completament de direcció i començar a treballar a una de les *Big Four*. Opcions completament diferents i extremadament interessants, però durant aquest procés va aparèixer l'oportunitat de fer un doctorat. No m'ho havia plantejat, però semblava encaixar perfectament. Crec que si el Dr. Jorge Lobo i a la Dra. Nava Rubin no m'haguessin ensenyat el món de la recerca universitària aquesta opció mai hagués estat viable.

En aquells moments tenia el grau en telecomunicacions i el màster en intel·ligència artificial, però una manca de coneixement sobre aviació. Després de rumiar-ho bastant vaig decidir aplicar el que els pares m'havien ensenyat: per intentar-ho mai es perd res. Així que vaig decidir aplicar a l'oferta de doctorat que oferia la UPC. A partir d'aquest moment va ser quan vaig conèixer a la Dra. Esther Salamí, amb la qual compartiria tres anys de coneixement i creixement personal. Moltes gràcies Esther pel temps i la dedicació. Gràcies per ajudar-me a ser un millor investigador. Aquesta tesi no hauria estat possible sense la teva ajuda. També voldria agrair a la Dra. Cristina Barrado per fer-me arribar l'oferta, ja que, pel meu compte segurament és una opció que no m'hagués plantejat. L'objectiu mai va ser que jo apliqués, però estic encantat d'haver-ho fet. Gràcies al Dr. Enric Pastor per també unir-se al projecte.

Gràcies a EUROCONTROL pel suport, l'ajuda, la guia i el finançament aportat. He de reconèixer que descobrir aquesta organització va ser una de les claus per a que jo ara estigui escrivint aquestes paraules. Segurament hi va haver més gent involucrada, però gràcies al Gilles Gawinowski per fer-ho possible i gràcies per l'ajuda a l'inici de la tesi en definir conceptes i objectius. També voldria agrair al Ramon Dalmau la càlida rebuda que em va proporcionar quan vaig anar a l'EUROCONTROL Innovation Hub i per compartir la feina que havia estat fent amb mi. Així com agrair a la Camille Anoraud el temps invertit a clarificar conceptes. La primera reunió a l'EUROCONTROL Innovation Hub quasi no la podem fer a causa de la COVID-19, però van aconseguir salvar-la. Sense aquesta primera reunió tot hauria estat molt més complicat.

També voldria donar les gràcies a la Dra. Esther Salamí i al Dr. Xavier Prats per oferir-me l'oportunitat de treballar en el Dispatcher3. La COVID-19 ho ha fet tot molt més complicat per culpa de l'aïllament, i sense el Dispatcher3 l'experiència del doctorat no hauria estat el mateix.

Relacionat amb el Dispatcher3, voldria agrair enormement el temps, l'esforç i el coneixement compartit del Dr. Luis Delgado. Ha estat un plaer treballar conjuntament. Espero que algun dia ens puguem conèixer en persona. També voldria deixar constància del temps invertit treballant amb el Dr. Paolino de Falco i en Georgy Skorobogatov. Tothom té alguna cosa a ensenyar.

De l'oferta a la tesi que el lector té ara mateix entre les mans han passat moltes coses, de bones i de dolentes, però orgullós d'haver superat tots els obstacles que m'he anat trobant. Les meves últimes paraules són per a la família. Tot i ser les últimes, no vol dir que siguin les menys importants, sinó el contrari. Mai hauria arribat a aquest punt sense els pares i l'Iciar al costat. Gràcies a la mare per un suport incondicional, per fer-ho possible i per continuar lluitant. Gràcies pare per lluitar fins a l'extenuació. I gràcies Iciar Puigpelat pel suport, l'ajuda, la paciència, la "xarleta" i els moments únics que he passat al teu costat. Continuarem buscant la X allà on anem. A tots vosaltres, moltes gràcies per tot i per ser com sou. Aquesta tesi també és en part vostra.

Barcelona, Castelldefels, Març de 2023
Sergi Mas-Pujol

# Resum

Actualment, els proveïdors de serveis de navegació aèria han de gestionar i acomodar una demanda de tràfic aeri en constant creixement en un escenari que s'espera que sigui més eficient en temps i costos. Ajustar la demanda a la capacitat de l'espai aeri disponible és un dels problemes més complexos als quals s'enfronta la gestió del trànsit aeri. Aquest procés col·laboratiu de gestió de la capacitat sovint acaba imposant regulacions quan la capacitat no es pot ajustar. Assignant retards a l'aeroport de sortida, el trànsit es distribueix i les arribades es regulen a la infraestructura congestionada. Tot i això, decidir on i quan es necessita una regulació requereix temps i es basa en gran manera en el coneixement i l'experiència. Això porta a regulacions subòptimes i innecessàriament llargues, cosa que es tradueix en retards innecessaris i una no òptima utilització de la capacitat.

Al llarg dels anys, molts investigadors han estudiat noves tècniques per estimar millor la complexitat d'un sector aeri – volum aeri – o com quantificar la càrrega de treball dels controladors aeris amb l'objectiu d'identificar les regulacions necessàries per a una correcta gestió del trànsit aeri. A causa del gran impacte que provoquen els retards a la xarxa, es poden trobar una gran varietat de treballs tractant d'optimitzar, millorar, minimitzar o predir l'evolució dels retards. La literatura mostra tres tendències principals: propostes sense intel·ligència artificial, enfocaments utilitzant aprenentatge automàtic supervisat o treballs explorant tècniques d'aprenentatge per reforç. Tot i això, hi ha una mancança de treballs que se centrin concretament en la identificació de les regulacions necessàries, i els mètodes proposats per suavitzar la demanda pateixen problemes d'escalabilitat.

La finalitat principal d'aquesta tesi és investigar l'ús de tècniques d'intel·ligència artificial per identificar i resoldre desequilibri entre la demanda i la capacitat que requereixen la implementació de regulacions durant la fase pretàctica. És a dir, quan no hi ha informació disponible de l'administrador de la xarxa sobre regulacions requerides i quan els nivells d'incertesa són molt més alts.

Primer, s'ha estudiat la identificació de regulacions a escala de sector aeri, fent servir tècniques supervisades i prototipant eines per l'entitat que gestiona l'espai aeri. S'estudien els dos tipus de regulacions més freqüents per a les regions més congestionades a Europa. Els resultats revelen que l'arquitectura proposada és capaç d'identificar gairebé totes les regulacions durant l'estiu, probablement la temporada més congestionada. Segon, s'investiguen tècniques d'aprenentatge per reforç en la resolució de les regulacions prèviament identificades, centrant-se en l'escalabilitat

del sistema gràcies a l'ús d'imatges. Finalment, s'investiguen els potencials beneficis d'identificar les regulacions a escala de vol. En aquest últim cas, els resultats també mostren que és possible predir les característiques de les regulacions fent servir tècniques supervisades. A més, la integració dels models permet avaluar l'impacte i la gravetat de les regulacions emeses, anticipant possibles retards reaccionaris.

En general, els resultats mostren que és possible predir amb precisió regulacions, les seves característiques i automatitzar el procés per suavitzar el tràfic quan es vol resoldre desequilibri entre la demanda i la capacitat. Hi ha alguns factors a tenir en compte que poden limitar els beneficis de les solucions proposades, començant pels problemes de disponibilitat de dades i el nombre d'estudis realitzats. No obstant això, les eines desenvolupades han estat provades en les regions europees més complexes. Finalment, desplegar les diferents eines desenvolupades seria clau per estudiar els beneficis i l'impacte de les solucions proposades. Per tant, s'han creat diferents eines per a la visualització dels resultats tenint en compte la incertesa de les solucions proporcionades.

# Abstract

Nowadays, Air Navigation Service Providers (ANSPs) have to handle and accommodate a continuously increasing traffic demand in a scenario that is expected to be more time-efficient and cost-efficient. Meeting the demand with the available airspace capacity is one of the most challenging problems faced by Air Traffic Management (ATM). This collaborative Demand-Capacity Balancing (DCB) process often ends up enforcing Air Traffic Flow Management (ATFM) regulations when capacity cannot be adjusted. The arrival traffic is spread out by assigning delays on the ground at the departure airport, and the arrivals are metered at the congested infrastructure. However, deciding whether and when regulations are needed is time-consuming and relies heavily on human knowledge. This leads to suboptimal and unnecessarily long regulation and, therefore, to the realization of unnecessary delay and underuse of the capacity.

Over the years, many researchers have investigated new techniques to estimate better the complexity of a given Air Traffic Control (ATC) sector – Traffic Volume (TV) – or how to quantify the workload of the Air Traffic Controllers (ATCOs) to identify required ATFM regulations. Moreover, because of the huge impact of ATFM delays in the network, a wide variety of previous work can be found trying to optimize, improve, minimize, or predict the evolution of delays. The literature shows three main trends: proposals without any Artificial Intelligence (AI), using supervised Machine Learning or Reinforcement Learning (RL) techniques. However, there is a lack of work directly focusing on the identification of required ATFM regulations and their characteristics, and the proposed methods to smooth demand-capacity imbalances suffer from scalability issues.

The main objective of this PhD thesis is to investigate the usage of AI techniques to identify and smooth DCB problems leading to ATFM regulations during the pre-tactical phase. That is when there is no available information from the Network Manager (NM) about required regulations and when levels of uncertainty are much higher. Different sets of frameworks are studied and developed, considering the needs and policies of different stakeholders.

First, it is studied the identification of ATFM regulations at the TV level, using supervised techniques and developing a framework that aims to be used by the NM. The two most frequent regulations reasons are analyzed over two of the most congested European Civil Aviation Conference (ECAC) regions. Results reveal that the proposed architecture can identify almost all the regulations during the summer, which is probably the most congested season. Second, RL techniques are investigated to solve the previously identified ATFM regulation, focusing on scalability due to the usage of images. Finally, airlines are the stakeholders affected by ATFM regulations;

thus, the potential benefits of identifying ATFM regulations at the flight level are also analyzed. Promising results show it is possible to predict ATFM characteristics using supervised techniques. Moreover, the models are integrated into a framework to assess the impact and severity of issued regulations to anticipate possible reactionary delays for specific aircraft frames.

Overall, results prove it is possible to accurately predict ATFM regulations, the characteristics of such regulations, and automatize the smoothing process required to solve DCB issues. There are some factors to be considered that may limit the benefits of the proposed solutions, starting with data availability issues in some experiments. However, it is worth mentioning that the models have been tested under the most challenging European scenarios. Finally, deploying the proposed framework will be key to studying the benefits and impact of the proposed solution. Therefore, specific advice capabilities are proposed for the visualization of the results taking into account uncertainty.

# Notation

Throughout this PhD thesis and as a general rule, scalars and vectors are denoted either with lower or upper case letters. Vectors are noted with the conventional overhead arrow, such as $\vec{a}$ or $\vec{\psi}$. Sets are denoted using calligraphic fonts, for instance, $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{X}$, while matrices use the same font but in bold series, like $\boldsymbol{\mathcal{R}}$. In the notation , $(\cdot)^*$ indicates *optimal*. Next, the principal symbols that are used throughout this dissertation are shown along with their meaning. The reader should note that this list is not exhaustive.

$C$ capacity traffic volume

$C_t$ cell state vector

$D_i$ ground delay imposed by agent $i$

$F$ Fahrenheit

$F_t$ forget gate

$G$ cumulative reward

$I(z)$ demand-capacity ratio

$I_t$ input gate

$K$ kelvin

$M$ number of delayed flight

$O_t$ output gate

$R_t$ reward at time step $t$

$S_t$ state vector time-step $t$

$T$ trajectory from flight plan

$TP$ time period

$T_X$ input time step X

$V_t$ expected occupancy count in a particular traffic volume

$\Theta(\mathcal{N})$ function that counts the number of flights that received ATFM delay

$\alpha$ learning rate

$\beta$ weight adjust penalty number flight delayed

$\delta$ weight adjust penalty delay

$\gamma$ discount factor

$\lambda$ weight adjust penalty ratio demand-capacity

$\mathbb{R}$ conjunt real numbers

$\mathcal{A}$ set of actions

$\mathcal{N}$ number of interacting agent

$\mathcal{O}$ observation state

$\mathcal{P}$ counting period

$\mathcal{P}$ transition function

$\mathcal{R}$ reward function

$\mathcal{S}$ state space

$\mathcal{T}$ set of trajectories

$\pi$ policy

$\sigma$ sigmoid function

$a_t^i$ action $i$ from set $\mathcal{A}$ at time-step $t$

$a_t$ action at time-step $t$

$f$ function to compute the distance between two coordinates

$h_t$ hidden state vector

$o_t$ observation at time step $t$

$q_\pi(s\,a)$ value of taking action $a$ in state $s$ under a policy $\pi$

$s$ seconds

$s'$ next state

$s_t^i$ state $i$ from set $\mathcal{S}$ at time-step $t$

$tahn$ tangent hyperbolic function

$v_pi(s)$ value function of a state

$v_pi(s)$ value function of a state

$v_{ID}$ expected velocity of the flight in a particular segment

$z$ represents the system under evaluation

$X$ regulated ATFM timestamp

# List of Acronyms

| | |
|---|---|
| ACC | Air traffic Control Center |
| ADP | ATFCM Daily Plan |
| AFP | Airspace Flow Program |
| AI | Artificial Intelligence |
| AIRAC | Aeronautical Information Regulation and Control |
| ANN | Artificial Neural Network |
| ANOVA | ANalysis Of VAriance |
| ANS | Air Navigation Service |
| ANSP | Air Navigation Service Provider |
| API | Application Programming Interface |
| ATC | Air Traffic Control |
| ATCO | Air Traffic Controller |
| ATFCM | Air Traffic Flow & Capacity Management |
| ATFM | Air Traffic Flow Management |
| ATM | Air Traffic Management |
| ATMAP | ATM Airport Performance |
| AU | Airspace User |
| AUC ROC | Area Under the Receiver Operating Characteristic Curve |
| AWS | Amazon Web Services |
| BPH | Best Predicted vs Human |
| CASA | Computer Assisted Slot Allocation |
| CDM | Collaborative Decision Making |
| CFMU | Central Flow Management Unit |
| CHMI | Collaboration Human Machine Interface |
| CNN | Convolutional Neural Network |
| CTOT | Calculated Take-Off Time |
| D-1 | day prior to operations |
| D0 | day of operation |
| DCB | Demand-Capacity Balancing |
| DD | Dynamic Density |
| DDPG | Deep Deterministic Policy Gradient |
| DDR | Demand Data Repository |
| DDR2 | Data Demand Repository 2 |
| DL | Deep Learning |
| DPG | Deterministic Policy Gradient |

| | |
|---|---|
| DQN | Deep Q-Learning |
| EATMN | European Air Traffic Management Network |
| EC | Entry Count |
| ECAC | European Civil Aviation Conference |
| ECMRWF | European Centre for Medium-Range Weather Forecasts |
| EDA | Exploratory Data Analysis |
| EDC | Equilibri de la Demanda i la Capacitat |
| EOBT | Estimated Off-Block Time |
| ETFMS | Enhanced Tactical Flow Management System |
| ETOT | Estimated Take-Off Time |
| FCM | Flow and Capacity Menagement |
| FFNN | Feed Forward Neural Network |
| FI | Flight Intention |
| FL | Flight Level |
| FMP | Flow Manager Position |
| FRA | Free Rounting |
| FTFM | Filed Tactical Flight Model |
| GATA | Gestió d'Afluència de Trànsit Aeri |
| GDP | Ground Delay Progam |
| GPU | Graphical Processing Unit |
| GRU | Gated Recurrent Units |
| GTA | Gestió del Trànsit Aeri |
| ICAO | International Civil Aviation Organization |
| IFR | Instrument Flight Rules |
| INAP | Integrated Network management & ATC Planning |
| KPI | Key Performance Indicators |
| LIME | Local Interpretable Model-Agnostic Explanations |
| LSTM | Long-Short Term Memory |
| MAA3C | Multi-Agent Asynchronous Advantage Actor-Critic |
| MAE | Mean Absolute Error |
| MARL | Multi-Agent Reinforcement Learning |
| MDP | Markov Decision Process |
| METAR | METeorological Aerodrome Report |
| ML | Machine Learning |
| MLP | MultiLayer Perceptron |
| MSE | Mean Squared Error |
| MUAC | Maastricht Upper Area Control Centre |
| MVC | Model-View-Controller |
| NAT | North Atlantic Traffic |
| NextGen | Next Genration Air Transport System |
| NM | Network Manager |
| NN | Neural Network |
| NOAA | National Oceanic and Atmospherics Administration |
| NOP | Network Operations Plan |
| NWP | Numerical Weather Prediction |
| OC | Occupancy Count |
| OD | Origin-Destination |
| PCA | Principal Component Analysis |
| PSNA | Proveïdors de Serveis de Navegació Aèria |
| ReLU | Rectified Linear Unit |
| RL | Reinforcement Learning |
| RMSE | Root Mean Squared Error |
| RNN | Recurrent Neural Network |
| SESAR | Single European Sky Air traffic management Research |
| SHAP | SHapley Additive exPlanations |

| | |
|---|---|
| SIMEX | SIMulation and EXperiment |
| SOBT | Scheduled Off-Block Time |
| TBO | Trajectory-Based Operations |
| TD | temporal-difference |
| TV | Traffic Volume |
| XAI | eXplainable Artificial Intelligence |

*One day we will be old*
*and think about all stories*
*that we could have told*

— Asaf Avidan

# I

# Introduction

Over the past 50 years, aviation has experienced a continuous and rapid expansion, with an underlying annual air traffic growth rate close to 5% since 1960 (Oxley & Chaitan, 2018). Air transport is an efficient, safe, and fast means of transport for cargo and passengers, contributing to world development and the economy. According to Air Transportation Action Group (2019), 65.5 million jobs (direct and indirect) were supported by aviation worldwide, carrying 4.1 billion passengers in 41.9 million commercial flights worldwide and flying 7.75 trillion passenger kilometers in 2019.

The Performance Review Commission (PRC, 2019) reported that 2018 was the fifth consecutive year of air traffic growth in the European Civil Aviation Conference (ECAC) area, with a 3.8% average increase in the Instrument Flight Rules (IFR) flights over 2017. The peak traffic load reached the highest level of traffic on September $7^{7th}$, when the system handled more than 37 thousand flights. However, such continued growth contributed to a further decrease in overall service quality, following the trend observed in past years. The number of flights arriving within 15 minutes of their scheduled time decreased by 3.9 points, reaching the lowest level in the last ten years.

At the time of writing this thesis (October 2022), air traffic has survived its second year of the COVID-19 pandemic. Vaccination and relaxation of travel restrictions in European resulted in a continuous increase in demand. However, the terrible war in Ukraine set back any hopes for a sustained and swift recovery from COVID-19 in early 2022. Despite the unprecedented drop in 2020, traffic in the ECAC area recovered 6.2 million flights in 2021, corresponding to roughly just over half of the traffic in 2019. In the base scenario forecasted by EUROCONTROL (STATFOR, 2022), it is expected a complete recovery by 2024 (see Figure I-1).

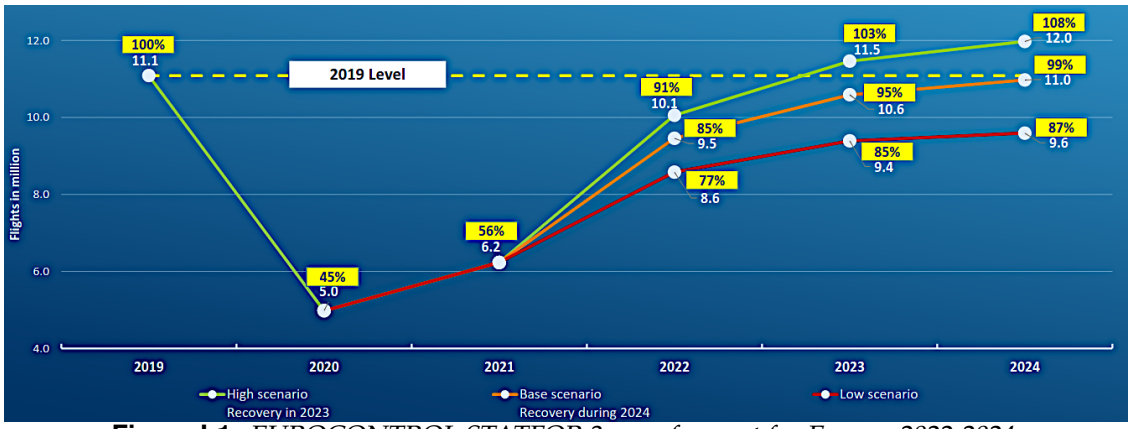



**Figure I-1:** *EUROCONTROL STATFOR 3-year forecast for Europe 2022-2024. Source: STATFOR (2022)*

The reduced traffic levels during the recovery phase presented a good opportunity to review and remove efficiency constraints in the Air Traffic Management (ATM) system, aiming to improve capacity planning and deployment to fit the increasing traffic levels during the recovery phase and beyond. However, in 2021 there were early indications of rising inefficiency despite traffic levels still being well below those of 2019 (PRC, 2021). Based on the latest capacity plans (STATFOR, 2022), a high-traffic scenario is expected with a delay of 1.78 minutes per flight, which is a reason for concern.

In view of the expected continued growth, all signs indicate that the delay situation will deteriorate drastically if bold actions are not taken. This so-called capacity crunch prediction indicates that airports will be unable to accommodate approximately 1.5 million flights in 2040, or around 160 million passengers (SESAR, 2020). In Europe, the Single European Sky Air traffic management Research (SESAR) program addresses the impact of air traffic growth by implementing novel procedures and technologies to increase airspace capacity and efficiency in the ATM system while simultaneously improving safety and reducing the environmental impact. One of the SESAR ambition is to enable a 5-10% capacity increment in highly congested areas (SESAR, 2020).

The issue is not the lack of overall capacity but rather the lack of capacity in specific locations or at certain temporal periods. On the other hand, from a time performance view, the ambition is to reduce the average departure delay per flight in the ECAC area from 9.5 to 8.5-6.5 minutes, which is expected to come from reactionary delays (-2.26 minutes), airports delays (-0.16 minutes), and en-route delays (-0.04 minutes) (SESAR, 2020). Reactionary delays are caused by the late arrival of aircraft, crew, passengers, or baggage from previous journeys. Airport delays are mainly related to boarding, baggage handling, aircraft cleaning, fuelling, catering, technical defects, and late crew boarding or crew shortage. En-route delay is typically linked to re-routing, holding, or congestion issues.

Meanwhile, in the United States, 21% of the flights were delayed more than 15 minutes in 2021, with 3.38% flights canceled or diverted (FAA, 2018). In this case, the American Next Genration Air Transport System (NextGen) program (FAA, 2022a) is addressing the impact of air traffic growth by developing and implementing novel procedures and technologies. Similarly, China also suffers a severe issue with flight delays, even though its flight demand accounts for only around 1/3 of United States demand with almost equal airspace size (Hsu, 2014).

## I.1 Demand-Capacity Balancing: Reaching the system capacity

In the coming years, Air Navigation Service Providers (ANSPs) will have to handle and accommodate a continuously increasing traffic demand in a scenario that is expected to be more time-efficient and cost-efficient. Therefore, the most challenging problem facing the ATM will be to meet the capacity of the airspace sectors with the growing demand, while the safety levels must be maintained or increased. To ensure the successful distribution of flights, ANSP start defining their capacities a year to six months prior to day of operation (D0). According to the planning phase (*i.e.*, strategic, pre-tactical, or tactical), maximum capacity values are typically estimated based on historical traffic levels, geometrical characteristics of the airspace, Air Traffic Controller (ATCO) workload models, staff availability, or weather conditions (Tobaruela *et al.*, 2013).

The process of ensuring that the demand is under the capacity is known as Demand-Capacity Balancing (DCB), and it is done by the Flow Manager Position (FMP) role. It starts months in advance regarding the day of operation, typically when the airlines and flight operators submit the initial flight intentions. It is a cyclic process that aims to ensure that no ATCO will have to manage an airspace sector where the air traffic demand is above a predefined threshold (capacity). Figure I-2 summarizes the main steps of the DCB process: monitoring, detection, and resolution.


**Figure I-2:** *Main steps to detect and smooth demand-capacity imbalances.*

DCB is a particularly complex problem. First, automatic tools will report the locations (place and time) where the expected demand exceeds the predefined capacity (imbalance). Second, the imbalance will be studied manually by the FMPs. Finally, the necessary actions are defined and implemented to smooth the demand if required. In the ECAC region, the DCB process is carried out in four phases (Niarchakou, 2022):

1. **Strategic flow management:** takes place seven days or more prior to the D0. This phase focuses on continuous data collection and the identification of major demand-capacity imbalances. The output of this phase is the Network Operations Plan (NOP);
2. **Pre-tactical flow management:** is applied during the six days prior to the D0. This phase compares the demand for the day of the operation with the predicted available capacity and makes any necessary adjustments to the NOP. The output is the ATFCM Daily Plan (ADP);
3. **Tactical flow management:** takes place on the D0 and involves considering, in real-time, those events that affect the ADP, making the necessary modifications to it. This phase aims to ensure that the measures taken during the previous phases are the minimum required;
4. **Post operational analysis:** measures, investigates, and reports on operational processes relevant to DCB measures. The outcome of this phase is the development of best practices to improve operational processes and activities.

From the previous phases, it can be seen that the DCB process has two steps: the **detection** of demand-capacity imbalances and the **resolution**. To guarantee the detection and to ensure that the proposed resolution is effective, two capacity values and an overload duration threshold are defined per metric and active airspace sector (Flynn *et al.*, 2003):

- **Maximum peak occupancy capacity** represents the maximum number of flights that can be handled in an airspace sector simultaneously. Typically, 20, 60, or 120 minutes counting periods are used;
- **Sustained occupancy capacity** corresponds to the acceptable number of flights that can be handled in an airspace sector under specific circumstances, and in particular, according to the counting period. Typical counting periods are 20, 60, or 120 minutes.

Active airspace sectors are those present in the defined sector configuration for the day of operation. This configuration is the decomposition of air traffic services to guarantee a manageable workload according to the available resources. In the European network, there are two types of sectors: elementary sectors are the minimal division of the airspace, and grouping different elementary sectors create collapsed sectors.

On the other hand, according to the time horizon to D0 (uncertainty, granularity, and predictability of the information), the FMP prioritizes some metrics over others to identify DCB issues. More generic metrics are considered in large time horizons with high uncertainty (e.g., aircraft density). However, when the time horizon is close to D0 with less uncertainty, accurate metrics such as the interactions between specific pairs of aircraft can be used. Figure I-3 shows, from a high-level point of view, the metrics used at different stages of the DCB process. The size of the circles and the color map are used to represent the uncertainty of the information.

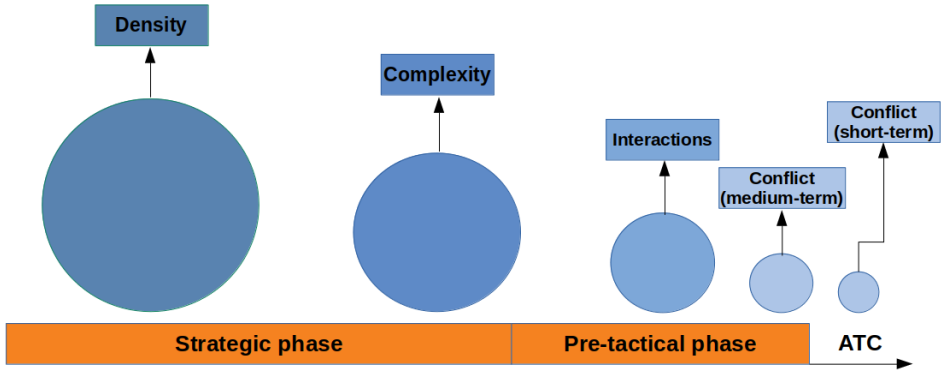



**Figure I-3:** *Metrics with respect to the temporal distance to the D0. The size and color map represent the uncertainty according to the time horizon.*

Initially, demand-capacity imbalances are solved via airspace management or flow management solutions (*e.g.*, STAM measures). However, when none of these solutions are enough, Air Traffic Flow Management (ATFM) regulations are implemented, issuing extra ground delay to the necessary flights (Bertsimas & Patterson, 1998). This cascade of events increases the uncertainty regarding the scheduling of operations, costs (Cook & Tanner, 2015), and unforeseen effects on the entire system. Furthermore, these events present further adverse effects for the ATM stakeholders, including loss of reliability, customer satisfaction, and environmental effects.

Airspace management solutions are based on changing the sector configuration and redistributing the available resources, aiming to reduce the expected workload. Sector configuration is challenging because it mixes a graph partition problem and an NP-hard optimization problem. Classical optimization of sector configuration has been widely studied (Xue, 2009), while over the years, solutions based on machine learning techniques have gained popularity (Gianazza & Alliot, 2002). On the other hand, flow management solution focus on redirecting air traffic flows to reduce part of the expected demand, using techniques such as re-routing or level-capping. Chan & Lin (2005) and Peeters *et al.* (2018) are examples showing the effects and effectiveness of re-routing and level-capping. Similarly, studies based on optimization can be found

in Samà *et al.* (2012) where an alternative graph formulation was presented based on a rolling windows framework, or Prats *et al.* (2018) highlighted the principal outcomes of the APACHE Project.

Current ATFM solutions are even more challenging because all required processes are done in the FMPs mind by building up a mental picture of the flights' intention (SESAR, 2019a). Moreover, the FMPs have to consider that, in realistic operations, a certain amount of capacity overloads are usually allowed (Melgosa *et al.*, 2019). Several reasons could explain this phenomenon: the lack of initial schedules for non-planed flights, the use of entry rates for assessing the demand without considering the occupancy, or a conservative approach for estimating the capacity and the complexity. Because of the complexity of the situations, vast amounts of information must be processed. As a summary, the authorities use as reference Table I-1 to determine the workload considered as overload.

**Table I-1:** *Overload threshold for the ATCOs. Source: Flynn* et al. *(2003)*

| Demand over capacity threshold | Interpretation | Working time during 1 hour |
|---|---|---|
| 70% or above | Overload | 42 minutes + |
| 54% - 69% | Heavy Load | 32 - 41 minutes |
| 30% - 53% | Medium Load | 18 - 31 minutes |
| 18% -29% | Light Load | 11 - 17 minutes |
| 0% - 17% | Very light Load | 0 - 10 minutes |

Over the years, many researchers have investigated new techniques to estimate better the complexity of a given Air Traffic Control (ATC) sector or how to quantify the workload of the ATCOs. In Kopardekar & Magyarits (2003), the authors presented a multi-year, multi-organizational research initiative related to measuring and predicting sector-level complexity using Dynamic Density (DD). Similarly, in Welch *et al.* (2007), the authors presented a model to quantify the workload impact using traffic density, sector geometry, flow direction, and air-to-air conflict rates. A different approach was proposed in Chatterji & Sridhar (2001), trying to predict the controllers' workload mainly focusing on cognitive factors (e.g., number of keystrokes) or focusing on physiological factors (e.g., heart rate or electrocardiogram). However, the authors realized there are better approaches than the physiological factors to measure the workload because an ATCO job is primarily cognitive and information-intensive rather than physical and labor-intense. Finally, Gianazza (2017) compared several Machine Learning (ML) methods and analyzed the vast majority of existing complexity metrics doing a Principal Component Analysis (PCA) (Abdi & Williams, 2010) to find that the most representative factors are related to *traffic characteristics*, which are:

- **Airspace volume**[1] of the considered ATC sector;
- **Number of aircraft** within the sector boundaries at time $t$;
- **Incoming traffic** flow within the next **15 minutes**;
- **Incoming traffic** flow within a **1 hour** time horizon;
- Average absolute **vertical speed** of the aircraft within the sector;
- Number of **speed vectors** interacting with an angle greater than 20 degrees.

[1] A traffic volume is related to a single geographical entity (either an aerodrome, a set of aerodromes, an airspace sector, or a point) and may consider all traffic passing through that entity or only specific flows.

As it can be seen, DCB is a time-consuming process that relies heavily on human knowledge. When capacity cannot be adjusted, and operational constraints are required, ANSPs through the FMP, and the Network Manager (NM) operators agree on the required ATFM regulations. These regulations shall smooth the demand over the overloaded part of the network, ensuring the available airspace capacity meets the traffic demand, delivering a safe and ordered flow of air traffic. With this aims, the Flow and Capacity Menagement (FCM) systems are the core of ATFM services provided by the NM which includes the following tools (Niarchakou, 2022):

- The **Enhanced Tactical Flow Management System (ETFMS)** compares traffic demand, regulates demand, and load against the capacity to assess possible imbalances in the airspace and allows the implementation of measures such as regulations;
- The **PREDICT** system compares forecasted traffic and capacity to evaluate the load situation for the following days (up to 6 days in advance), using a rather simple approach to predict the flight plans that are not yet in the system. ATFM measures may be implemented in this system to assess their impact before being applied;
- **SIMulation and EXperiment (SIMEX)** is used in strategic, pre-tactical, and tactical ATFM operations. It enables Network Operations staff to simulate ATFM measures or restrictions before applying them to the previous systems;
- A functionality called **OPTI-mise CON-figuration (OPTICON)** helps in the choice of sector configuration and enables better assessment of the impact of the change of configuration;
- The **Data Distribution System (DDS)** is used to distribute real-time flight data to the stakeholders involved;
- The **NOP Portal** is an interface that provides a consolidated view of the different aspects of the NOP and gives access to a set of services to support the NOP preparation and dissemination activities;
- The **Collaboration Human Machine Interface (CHMI)** is an application that allows users to display data (such as information on regulations or flight lists) and graphical information (such as routes, route attributes, airspace, and flight plan tracks) via map displays. This real-time information enables CDM between all partners.

Even though the huge variety of metrics, tools, and systems used during the collaborative DCB process, the methodology for deciding the configuration of the sector (opening schema) and required operational constraints (ATFM regulations) is purely human and does not rely on automation. However, in the SESAR 2020 Exploratory Research program, some projects have tried to improve the processes behind DCB. COTTON (Capacity Optimisation in TrajecTory-based OperatioNs) (COTON, 2018) project explored how the uncertainties associated with the agreed trajectory impact the quality of the predictions, the volume and complexity of traffic demand, and the effectiveness of DCB processes regarding airspace management. ISOBAR (Artificial Intelligence Solutions to Meteo-Based DCB Imbalances for Network Operations Planning) project (ISOBAR, 2020) investigated enhanced convective weather forecasts for predicting imbalances between capacity and demand. DART (Data-driven aircraft trajectory prediction research) project (DART, 2019) explored the potential of data-driven techniques for trajectory prediction and agent-based modeling approaches for assessing the impact of traffic on individual trajectories.

Notice that in this document, the term FMP is used to refer to the specialized ATCO whose main tasks focus on DCB activities. The reason is that it is a well-established acronym in the European community. However, Air Traffic Flow & Capacity Management (ATFCM) was the first acronym established, and the most updated name is Integrated Network management & ATC Planning (INAP).

## I.2 Air Traffic Flow Management Regulation: A power for good

In most European airports, to handle the excess demand, a finite number of slots are provided to airlines to schedule their flights. A slot is a permission to use the airport infrastructure and services to operate a flight with the purpose of landing or take-off. Therefore, the number of available slots per interval of time defines the capacity of the airport. However, when those slots are insufficient to meet the demand, the extra demand will generate air delay at the destination airport in the form of holdings or re-routing during the cruise (Delgado Muñoz, 2013). In the European ATM system, ground delay, rather than airborne delay, is widely accepted because of the reduced operational costs and environmental footprint (Cook *et al.*, 2004).

ATFM delays, which refers to the difference between the actual time the aircraft departed (or arrived) at the parking stand and the commercial schedule shared with the passengers, are particularly complex. First, when a flight is affected by an ATFM regulation, they are issued with a Calculated Take-Off Time (CTOT), which indicates a narrower time window for the flight to depart (from 5 minutes prior CTOT to 10 minutes after). If a flight cannot depart within this window, e.g., due to other delays, the ATFM slot will be missed and a new one assigned. This could lead to significant extra delays being issued to the airline as early slots might not be available. Therefore, CTOTs act as barriers when planning flights. Notice that if the delay is propagated and the ATFM slots are missed, this might have a significant downstream impact even if the initially assigned delay by the regulation is close or even zero. Airlines need to closely monitor if slots might be missed and notify the NM as soon as possible to obtain a new CTOT as close as possible to their new Estimated Take-Off Time (ETOT). On the contrary, if the initial delay is large, it can be used to absorb the propagation of delay from previous legs. Even if the flight is ready, it will not be able to depart until its CTOT window.

Second, in some cases, airlines can respond to the ATFM regulations. For example, if the regulation issuing the delay is in the airspace, a new flight plan which avoids the congested airspace (re-routing or maintaining a lower altitude) can be used to avoid entering that portion of the airspace, reducing (or eliminating) the issued delay. Moreover, suppose the aircraft is ready (crew and passengers boarded). In that case, messages can be exchanged with the NM to try to benefit from potential new early slots generated due to delays or cancellations by other flights. Within all causes, airlines tend to put their operational focus on arrival rather than departure punctuality. Late arrivals may cause passengers to miss their connecting flight, it causes reactionary flight delays, and under EU law, very late arrival may trigger financial compensation to passengers (EU regulation 261/2004) (EUROCONTROL, 2019).

In the European ATC network, ATFM delays are imposed by the Computer Assisted Slot Allocation (CASA) algorithm (Niarchakou, 2022; Tibichte & Dalichampt, 2014), which is a heuristic algorithm based on the principle first-planned-first-served. A similar program called the Airspace Flow Program (AFP) or the Ground Delay Progam (GDP) has been used in the United States of America since 1998. While CASA can apply delays to a subset of flights predicted to cross a regulated region, AFP or GDP only can delay a set of flights destined for a specific airport. For further comparison of procedures between Europe and the US, the reader is referred to Shetty *et al.* (2017).

According to PRC (2019), in the ECAC area ATFM delay increased by 104% in 2018, reaching 19M minutes (36.1 years), while traffic increased by 3.8% over the same period with 6.4M minutes of airport delay. Figure I-4 shows the total minutes of ATFM (y-axis) as a function of the number of ATFM regulated flights (x-axis). As can be seen, airport ATFM delays have stayed at a similar level over the last four years before COVID-19. In contrast, en-route ATFM delay presents continuous growth, especially in 2018. The most regulated locations in 2018 were Karlsruhe (21.3%), Marseille (15.2%), Maastricht UAC (7.8%), Reims (6.7%), Brest (5.4%), Vienna (4.3%) and Barcelona (3.8%). In 2021, the most constraining locations were Reims (18.5%), Marseille (15.2%), Karlsruhe (15.0%), and Athens (13.6%) (PRC, 2021).

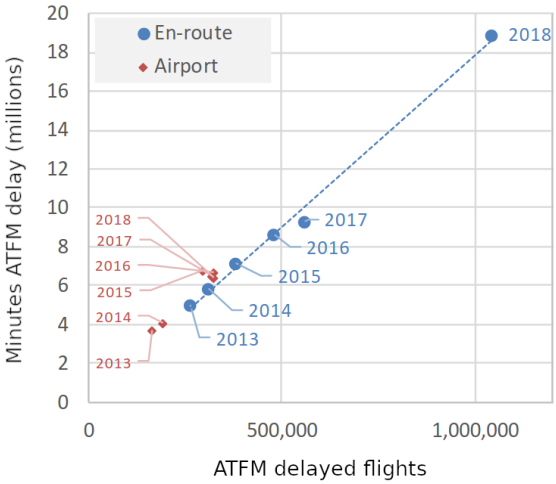



**Figure I-4:** *Airport and en-route minutes of ATFM delay. Source: PRC (2019)*

Because of the enormous impact of ATFM delays in the network, a wide variety of work can be found in the literature trying to optimize, improve, minimize, or predict the evolution of delays. For instance, Ivanov *et al.* (2017) presented an optimization model for the en-route demand-capacity imbalances to improve airport slot adherence, Ruiz *et al.* (2019) presented an innovative technique to utilize better available airspace capacities based on the current tools Prats Menéndez *et al.* (2017) presented an embedded simulator based on speed reduction, which combines simulation, optimization, and performance assessment, and Dalmau *et al.* (2021b) tried to estimate the evolution of already assigned ATFM delays.

Up to this point, it can be interpreted that ATFM regulations are purely traffic-related, but they can be implemented due to many circumstances. For instance, under adverse weather scenarios, the workload of ATCOs also rises significantly, mainly because the air traffic becomes irregular, difficult to anticipate, and there is less available airspace for conflict resolution. This increase in controllers' workload translates into a reduction of airspace capacity. Similarly, we can find regulation due to insufficient staffing, inefficient procedures , or inadequate equipment that cannot cope with the growth of air traffic. Another common reason is general disruptions, which mainly refer to industrial actions. Table I-2 shows the percentage of delayed flights per ATFM delay code in 2018. The complete list ATFM of delay codes can be found in EUROCONTROL (2022).

**Table I-2:** *Percentatge ATFM regulations codes for flights delayed in 2018. Source: PRC (2019)*

| ATFM Delay code | Delayed flights | Delay per delayed flight | Total delay |
|---|---|---|---|
| C - ATC Capacity | 4.3 % | 15 mins. | 37.4 % |
| S - ATC Staffing | 2.3 % | 17.5 mins. | 23 % |
| (W, D) - Weather | 1.9 % | 23.4 mins. | 25.4 % |
| (I, T) - ATC Disruptions | 0.4 % | 32.5 mins. | 7.5 % |
| All other codes | 0.6 % | 17.8 mins. | 6.6 % |

Traffic growth and changes in traffic patterns have caused increasing congestion and delay in European airspace. Both the SESAR program and the Central Flow Management Unit (CFMU) continually seeks and develops methods to improve traffic flow management to reduce delays and congestion (Tibichte & Dalichampt, 2014).

## I.3 On-going ATM paradigm shift

Digital transformation is not a goal in itself but a means of accelerating the SESAR ambitions. A digitally transformed aviation will use targeted data and information through automated and connected solutions to improve the overall efficiency and cost perspective (SESAR, 2020). SESAR aims to take full advantage of digital technologies to generate new services and optimize current ones while delivering the best possible experience and benefits to the different stakeholders.

SESAR (2020) introduced an automated model for ATC, which emulates the five-level model introduced by the Society of Automotive Engineers (SAE, 2018). Figure I-5 shows the different defined levels of automation, where it can be seen that the ATM network aims to take benefit from the progress made in the field of ML and Artificial Intelligence (AI). However, notice that it is not expected level 5 of automation because phase D only reaches level 4. The goal is a system that works collaboratively with hybrid human-machine teams, where flexible and adaptive automation could guide the tasks. Moreover, the synchronization between the air and the ground automation systems will make it possible to reduce the ATCO workload, thus reducing the required stuff or increasing capacity. Similar intentions exist in the United States, within the NextGen program, aiming to modernize the US air transportation system (FAA, 2016).

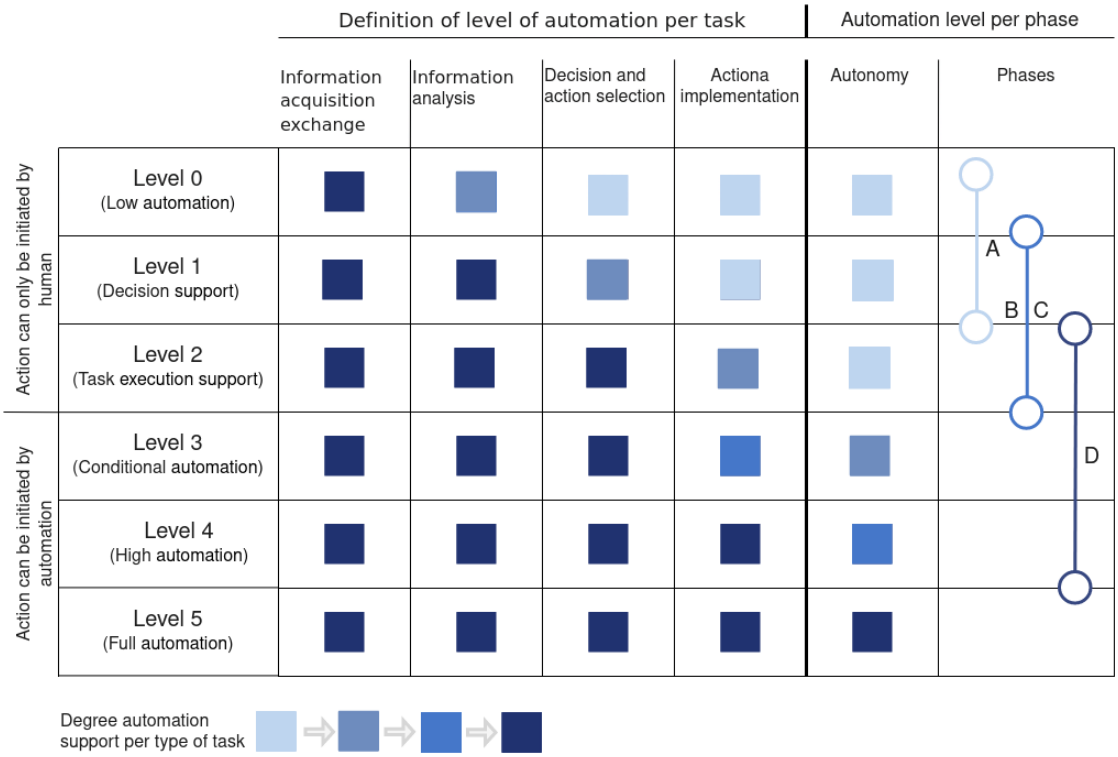



**Figure I-5:** *ATM levels of automation. Source: SESAR (2020)*

From level *0* to *2*, ATC and ATFM automation will focus on increasing the level of system support while the human is still responsible for initiating the actions. In level *3*, automation between humans and machines will be enabled, and automation can initiate actions for some specific tasks. Level *4* starts to remove the human from the loop that selects ATC actions; thus, the human cognitive limitations will no longer limit the capacity of airspace by design (SESAR, 2020). Finally, automation will perform all the tasks in level *5*. The boundaries between ATC and ATFM will progressively blur when automation makes possible the implementation of more flexible ATFM concepts. The most updated European ATM Master Plan from SESAR (SESAR, 2022) shows that automation Level 3 is under development with 130 solutions.

Most of the proposed solutions, mainly those more advanced, rely on the notion of Trajectory-Based Operations (TBO) enabling airspace users to fly their preferred flight trajectories. The trajectory will be defined before departure, updated in response to emerging conditions and operator inputs, and shared between stakeholders and systems. The aggregate aircraft trajectories on the day of operation will define demand and inform traffic management actions. This free routing concept enables airspace users to fly as close as possible to their preferred trajectory without being constrained by fixed airspace structures or fixed route networks (FAA, 2022b; EUROCONTROL, 2022). Concretely, in Europe, the concept of TBO and Free Rounting (FRA) is under analysis and validation in the project SESAR (2019b), with promising results on the sector's efficiency, capacity, and environmental problems.

## I.4 Motivation of this PhD thesis

In the current system, ATFM regulations are implemented when demand-capacity imbalances cannot be solved using airspace or flow management solution (STAM measures). Although they are the last option when solving DCB problems, they are widely used in the ECAC network. Moreover, the downstream effects of such regulations cannot be neglected. The average cost derived from ATFM delay per minute is €100; that is, around €1 billion annually at the European level (EUROCONTROL, 2020; Cook & Tanner, 2015). However, as presented in SESAR (2019a), the methodology used nowadays is purely human and does not rely on automation. The decision-making process is done by building a mental picture of flights intent in the mind of the controllers. For this reason, both the SESAR (Europe) and the NextGen (United States) programs aim to develop a new system that performs collaboratively by hybrid human-machine teams. This approach aims to reduce the workload of humans, allowing machines to do repetitive tasks more efficiently and cost-effectively. Thus, increasing capacity or reducing ATFM delay.

FMPs are typically conservative with ATFM regulations, preventing costly airborne holding and maximizing safety (Dalmau *et al.*, 2021a). Therefore, ATFM regulations are usually planned to last longer than necessary. It is preferable to have flights held on the ground even in unnecessary situations and cancel the ATFM regulations earlier if possible to release some demand. In 2018, around 5% of the regulations were canceled. Therefore, a more automated system able to identify regions and intervals of time which will have to be regulated could reduce the workload from FMPs and ATCO, increasing the overall capacity of the network. Moreover, current techniques can help smooth the traffic of congested regions better, taking into account different Key Performance Indicators (KPI) rather than only considering the sequence of flights. On the other hand, airlines need to closely monitor flights to mitigate ATFM delays and actively produce new flight plans and solutions to reduce the impact of delays on their fleet. Not only if a flight is impacted by ATFM delay, but the characteristics of this (amount of delay and type of regulation) are required as soon as possible for effective fleet management.

Such automation detecting and solving ATFM regulations could heavily impact the overall performance of the ATM network towards the goal from the SESAR program to increase the capacity by 5-10% in high congested regions. An automatic system to suggest the characteristics of possible ATFM regulations and possible approaches to smooth the traffic perfectly fits in Levels 3 and 4 from SESAR (2020). For these levels, automation supports the human operator in the information acquisition and exchange, information analysis, action selection, and action implementation for some tasks. Moreover, improved automation of the decision-making process behind the implementation of ATFM regulations could improve other related issues, such as the propagation of reactionary delays (De Falco & Delgado, 2021).

Last but not least, it should be mentioned that this thesis has been done in collaboration with EUROCONTROL, under PhD research Contract No. 18-220569-C2, who aimed to study the usage

of ML techniques to identify en-route ATFM regulations. Furthermore, part of this thesis belongs to the exploratory research project Dispatcher3 (Dispatcher3 Consortium, 2020), a Clean Sky 2 innovation action that aims to use machine learning techniques to support the airline processes prior to departure.

## I.5 Objectives of this PhD thesis

Automating the decision-making process behind ATFM regulations opens the possibility of new strategies for dealing with demand-capacity imbalances. As has been mentioned, it could increase overall capacity due to the reduction in traffic complexity, improve current capacity usage, and early identification of disruptions can enhance linked operations. This thesis focuses on the usage of **AI techniques for both the identification of DCB issues leading to ATFM regulations and the generation of advice on their resolution**. All this is in the context of SESAR and NextGen programs, where a higher level of automation is expected. Figure I-6 depicts a block diagram for the proposed framework, where discontinued lines are related to future work.

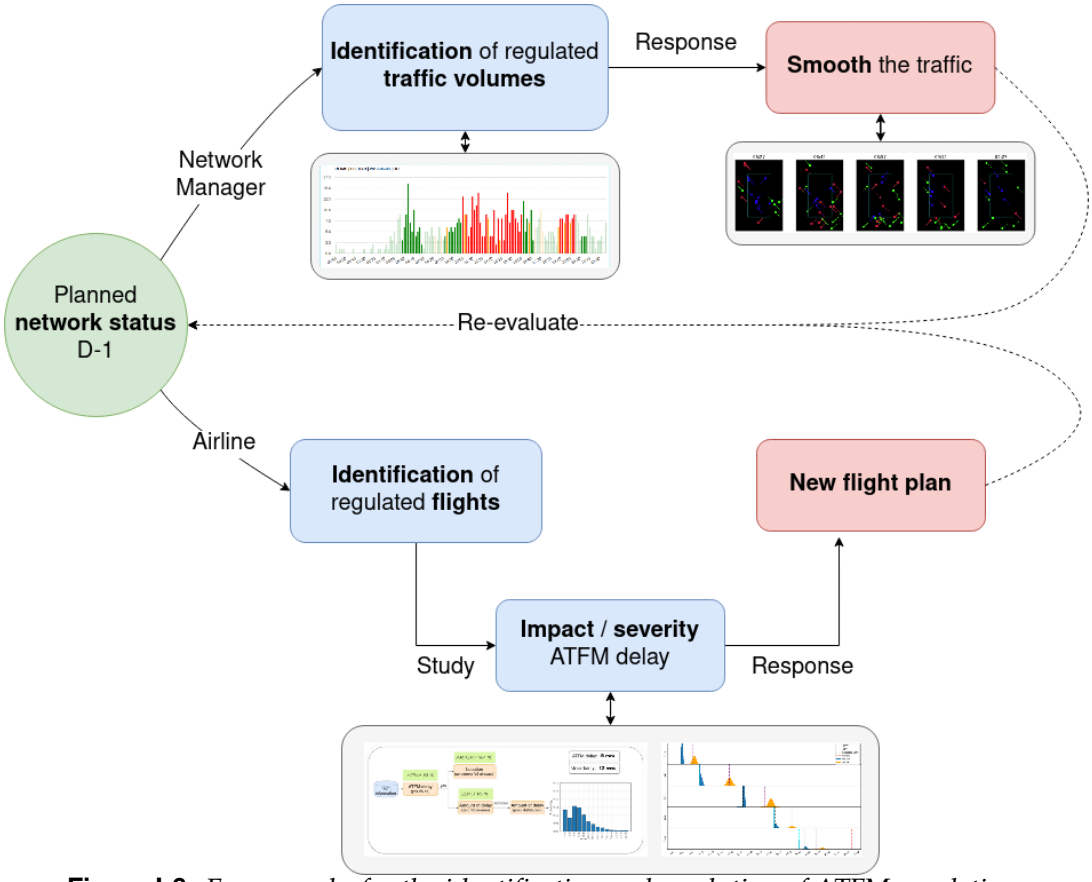



**Figure I-6:** *Frameworks for the identification and resolution of ATFM regulations*

*Supervised machine learning* and *Reinforcement Learning (RL) techniques* are analyzed, creating an end-to-end system to identify and suggest solutions for DCB issues leading to ATFM regulation. Different techniques are analyzed and compared to obtain the best possible performance, with an extensive analysis of possible input features for each case study, focusing on the interpretability and explainability of the results. *Supervised machine learning* is used to accurately identify where and when ATFM regulations are required, while *RL techniques* aim to generate advice about how to smooth traffic in the identified congested regions.

ATFM regulations in Europe are imposed by ANSPs and the FMPs, delaying flights on the ground to smooth demand. Thus, affecting the planned operations of the airlines. Because the operational and economic impact is different depending on the stakeholder, distinct approaches are analyzed to develop tools that fit the necessities of the different partners. First, a set of tools is developed to identify intervals of time when specific Traffic Volumes (TVs) will be regulated, meeting the necessities and policies of the NM. In this context, it is analyzed the two most frequent regulations reasons. Second, RL algorithms are tested using discrete and continuous actions to suggest ground delays to smooth traffic. Third, a set of models that predict ATFM characteristics for individual flights is studied, taking into account airspace users' needs.

AI techniques are the perfect approach due to their fast response to new conditions, making them ideal for cyclic processes such as DCB. However, some of them are considered "black boxes" which is unacceptable in critical environments. Therefore, special attention has been paid to model explainability to obtain theoretical guarantees on the expected behavior of machine learning-based systems during operation. Understanding the reasons behind the outcome of the models is crucial in assessing trust when we want to take action based on the outcome of the models. For this purpose, SHapley Additive exPlanations (SHAP) is the main tool selected.

Finally, it is paramount to ensure that the models provide meaningful advice, ensuring that the right level of information is displayed at every moment. Different advice capabilities to process the outcome of the predictive engines and transform them into actionable indications are considered taking into account stakeholders' policies. It is proposed a web application and integration into R-NEST for the NM, an integrated view and possible reactionary delay for the airspace users, and an image-based representation for the resolution of detected imbalances.

Summing up, the objectives of this PhD thesis can be outlined as follows:

- Build a **data infrastructure**, emulating a data lake, able to accommodate, store, and retrieve the required data sources to conduct the different experiments. Moreover, the infrastructure has to retrieve the complex input features used during the training phase of the models;
- **Develop, train, and test** different machine learning models to predict ATFM information taking into account the needs of different stakeholders: regulated TV for the Network Manager and ATFM characteristics for airspace users;
- **Analyze, compare, and evaluate** different techniques, approaches, and implementations to figure out the best configuration for each experiment, and their hyper-parameters[2];
- Use **eXplainable Artificial Intelligence (XAI) techniques** to study the patterns learned by the supervised models and validate that their behavior is realistic, using mainly SHAP.
- Developed an **advice framework** per expected end user to provide meaningful information, focusing on interpretability. Combination and integration of the different machine learning models to display appropriate advice;
- Investigate different **RL techniques** to create advice on the resolution of demand-capacity imbalances, focusing on scalability. Study different types of actions and configurations.

[2]In machine learning, a hyper-parameter is a parameter whose value is used to control the learning process.

## I.6 Scope and limitations of this PhD

In order to accomplish the objectives of this PhD thesis, the research is subject to assumptions that define the scope:

- In the research undertaken in this PhD thesis, it is considered that the ATFM detection and resolution strategies are conducted during the pre-tactical phase, around 24 hours before the take-off time. The flight intentions, the Estimated Off-Block Time (EOBT), and routes used are based on planned intentions before any regulation was applied. In each experiment, it is used the closest available information to the prediction horizon (*i.e.*, to the pre-tactical planning phase);
- Real airspace and air traffic data from 2018 is used in all the experiments. Current data is not adequate due to the current breakdown situation due to COVID-19 and the war in Ukraine;
- To develop systems as fair as possible for all the stakeholders, there is only one type of flight, *i.e.*, there is no aircraft with priority. Furthermore, extra costs, and environmental impact, due to ATFM regulations are not considered;
- It is assumed that all the ATFM regulations implemented by the NM were correct and necessary, ignoring those ATFM regulations canceled by the NM;
- The detection of ATFM regulations for specific TVs (*i.e.*, at the network level) is centered on the Maastricht Upper Area Control Centre (MUAC) and REIMS regions, which contain the most crowded airspace sectors in Europe, ensuring that the evaluation is carried out in challenging scenarios. However, the proposed approach is generic enough to consider other airspace regions by training new models;
- For the detection of ATFM regulations at the flight level, the models are developed focusing on flights operated by Vueling, one of the most active operators in Europe. This is necessary due to computational constraints derived from trying to use the entire airspace traffic in the ECAC area. However, Europe has a very structured and regulated ATC system with little freedom to apply different policies, which indicates that the models could be re-trained for other airlines due to similar patterns of behavior;
- Pre-tactical routes are defined by sets of segments composed of the initial time, the initial coordinates, the final time, the final coordinates, and the flight level. Because the exact time and location are only known at the beginning and end of the segments, constant speed and flight level are assumed per segment. The reader is referred to Basora *et al.* (2017) for a similar approach;
- Supervised machine learning models learn patterns from historical information and use them in future predictions. In that sense, the models do not improve the decisions made in the past. However, the models also have the ability to improve learning as the quantity and quality of data increases, and thus it will benefit as better quality data become available or improved ATFM algorithms emerge;
- For the part of this thesis that focuses on the resolution of ATFM regulations, it is assumed that the airspace sectors with demand-capacity imbalance are known (interval of time with overload and location). The strategic sector's capacity is known, and rectangles can be used to approximate the shape of the sector to reduce the implementation complexity in this preliminary study;
- Many advice capabilities could be implemented according to the needs and policies of the different studied stakeholders. However, this thesis proposes to develop visualization frameworks as similar as possible to the current tools available.

## I.7 Outline of this PhD thesis

The present document is organized into seven Chapters and two Appendices, which are summarized as follows. It is worth noting that a broad state of the art of the main topics addressed in this PhD thesis has been presented before. A deeper and more specific review of the state of the art for each individual topic is included at the beginning of each Chapter.

- **Chapter** II presents a detailed framework for ATFM regulations, including a description of the main ML algorithms used, the proposed architecture, infrastructure, tools, pre-tactical data sources, the performance evaluation metric used and developed, the selected technique for model explainability, and the proposed advice capabilities of the frameworks according to the end user;
- **Chapter** III investigates the usage of supervised machine learning techniques to identify the need for C-ATC Capacity ATFM regulations at the TV level to support the NM processes. This Chapter evaluates the performance of models that use different types of input features;
- **Chapter** IV extends the analysis conducted in the previous Chapter by adapting the proposed architecture to identify W-Weather ATFM regulations. The goal is to validate whether the proposed architecture can be used to predict different ATFM regulation reasons;
- **Chapter** V analysis the use of RL techniques to provide advice on how to smooth the traffic of regulated TVs. Different algorithms with different configurations are tested, using images to overcome scalability issues identified in the state-of-the-art;
- **Chapter** VI focuses on developing ML models to predict ATFM information at the flight level to support the airline pre-departure processes during the operational plan definition phase. It is studied how feasible it is to predict the probability of ATFM regulation, the expected protected location, whether the ATFM delay is going to be zero, and the ATFM delay for flights operated by a specific airline;
- **Chapter** VII gives the conclusions that are drawn from this work and point out some future research that could be done in the direction of the presented research;
- **Appendix** A extends the results presented in Chapter III and Chapter IV when identifying C-ATC Capacity and W-Weather ATFM regulation, focusing on a new airspace region. Concretely, the results are centered on Spain, a highly regulated region but less than the ones used in the main experiments;
- **Appendix** B shows the results obtained using different data sources to predict ATFM regulations at the flight level. It compares the performance of the models using data from forecasts (Chapter VI) and optimal/perfect pre-tactical information, studying the impact of the sources with respect to the time horizon day prior to operations (D-1).

# II

# Framework on ATFM regulations

Air Traffic Flow Management (ATFM) regulations are a complex task that require the coordination of Flow Manager Positions (FMPs) and Network Manager (NM) to identify and smooth demand-capacity imbalances. The current Collaborative Decision Making (CDM) process behind this task is complex and well-tested, but there is a lack of automation. It is mainly based on human skills and knowledge; thus, Air Traffic Management (ATM) aims to move to an environment with higher levels of information sharing, where humans and machines can collaborate to improve the current environment.

As presented in the previous Chapter, this thesis aims to study the usage of Artificial Intelligence (AI) systems that could improve the current approach, for instance, automatizing the detection of ATFM regulations, indirectly increasing capacity, or reducing the number of required ATFM regulations that have to be manually studied.

This Chapter presents the framework developed and proposed to use Machine Learning (ML) techniques in the detection and resolution of ATFM regulations, taking into account the needs of different stakeholders. The *detection* aims to identify when are where ATFM will be necessary, while *resolutions* refers to smoothing the traffic in the identified congested region. Section II.1 introduces the supervised ML models that are proposed to use and, if desired, provides further references to the reader. Section II.2 details the architecture and approach developed. Section II.3 shows the infrastructure and tools required to conduct the different experiments. Section II.4 details the characteristics of the data sources studied. Section II.5 presents the performance evaluation metrics used across experiments and case studies. Section II.6 introduces the tool used for the model explainability analysis conducted to understand the behavior of the models. Section II.7 details the proposed advice capabilities of the different tools.

## II.1 Machine learning techniques

Machine learning, or automated learning, is a branch of artificial intelligence that allows machines to learn without being explicitly programmed for this specific purpose, creating systems that are not only smart but autonomous and capable of identifying patterns in the data to convert them into predictions. This technology is currently present in an endless number of applications.

Back in the 19$^{th}$ century, the first algorithms were presented trying to emulate the human brain neural network's biology to attempt to create the first intelligent machines. However, the panorama started to change at the end of the 20$^{th}$ century when massive volumes of data started to be available to train models, and computing power grew significantly.

Machine learning approaches are traditionally divided into three general categories according to learning paradigms:

- **Supervised learning**: The algorithm learns from a set of observations (*i.e.*, input examples) and their corresponding labels, and the goal is to learn a function that maps the inputs and outputs to make predictions of observations not seen during the training;
- **Unsupervised learning**: No labels are given to the learning algorithm, having to find the structure in its inputs on its own;
- **Reinforcement learning**: An agent, or multiple agents, interacts with a dynamic environment in which they have to perform a certain task (or several). The program provides feedback analogous to the reward, which agents try to maximize.

This thesis focuses on *supervised* and *reinforcement* learning techniques because the end goal is to learn from past scenarios to build up a system able to support the current process behind the Demand-Capacity Balancing (DCB).

### II.1.1 Supervised machine learning

Supervised learning algorithms build mathematical models from a training dataset that contains both the inputs and the labels. Each training observation has one or more input elements/features, typically represented by an array or vector, and the desired output. Through an iterative optimization process, supervised learning algorithms learn a function that can be used to predict the output of new inputs that were not used during the training process. Algorithms that improve the accuracy of their predictions over time are said or considered to have learned to perform that specific task (Mitchell, 1997). Note the goal is to generalize, not to memorize the training observations.

Types of supervised-learning algorithms include:

- **Classification** algorithms are used when the outputs are restricted to a limited set of values/categories;
- **Regression** algorithms when the outputs can have any numerical value within a range;
- **Similarity** learning uses a similarity function to learn how similar or related two objects are;
- **Active learning** is a case of machine learning in which a learning algorithm can query a user interactively.

Various types of models have been used and studied for supervised machine learning systems. The following list summarizes the ones used in this thesis:

- **Feed Forward Neural Networks (FFNNs)** are vaguely inspired by the biological neural networks that constitute animal brains, and they are also known as MultiLayer Perceptrons (MLPs). It is a model based on a collection of connected units or nodes called "artificial neurons", where each connection can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. The signal of connections between artificial neurons are real numbers, and the outputs are computed by non-linear functions of the sum of their inputs. Typically, neurons are aggregated into layers, where connections between neurons have a weight to increase or decrease the strength of a signal. Different layers may perform different kinds of transformation to their inputs, starting from the first layer (the input layer) to the last layer (the output layer), possibly after crossing multiple intermediate layers (the hidden layers). Figure II-1 shows the typical architecture of a MLP.

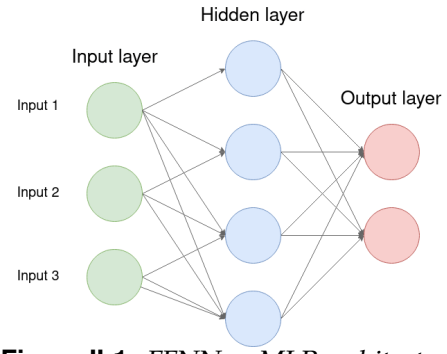



**Figure II-1:** *FFNN or MLP architecture*

- **Convolutional Neural Networks (CNNs)** are a class of ANN commonly used to process and analyze visual imagery based on the convolution of kernels, or filters, that slide along the input features to provide translation-equivariant responses known as feature maps. CNNs are regularized[1] version of MLPs. Figure II-2 presents the layers of conventional CNN.

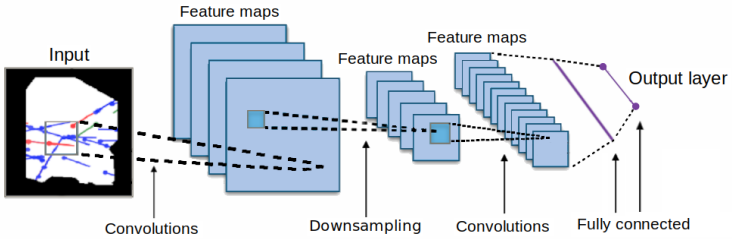



**Figure II-2:** *CNN architecture*

- **Recurrent Neural Networks (RNNs)** are a class of Artificial Neural Network (ANN) where information travels in loops from layer to layer so that the state of the model is influenced by its previous states allowing it to exhibit temporal dynamics. Long-Short Term Memory (LSTM) is a type of RNN widely used for sequence classification problems (Hochreiter & Schmidhuber, 1997), typically showing better performance than Gated Recurrent Units (GRU) or pure RNN. A common LSTM unit is composed of an input gate ($I_t$), an output gate ($O_t$), and a forget gate ($F_t$). The cell remembers values over arbitrary timestamps, and the three gates are used to regulate the flow of information into and out of the cell. Figure II-3 is a graphical example of an LSTM cell, showing the connectivity between elements and the mentioned gates.

[1]Regularization is a process that changes the resulting answer to be "simpler". It is often used to obtain results for ill-posed problems or to prevent overfitting.

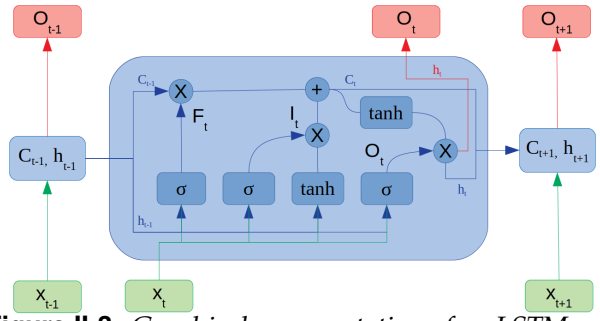



**Figure II-3:** *Graphical representation of an LSTM unit*

- **Decision trees** uses a "tree" as a predictive model to go from the input observations (represented as the branches) to the prediction (represented by the leaves). In other words, branches represent conjunctions of features, and the leaves represent the class labels. Decision trees where the target variable can take continuous values are typically called regression trees. Figure II-4 depicts an example of the branches and leaves of a decision tree algorithm.

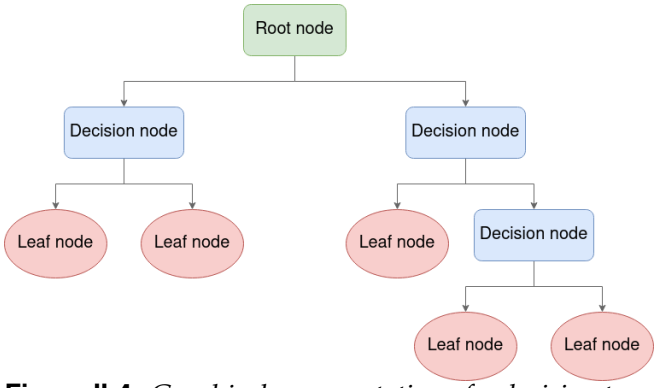



**Figure II-4:** *Graphical representation of a decision tree.*

- **Random forest** is an ensemble learning method for classification, regression, and other tasks that operates by constructing many decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Figure II-5 is a visual representation of a random forest composed of three trees.

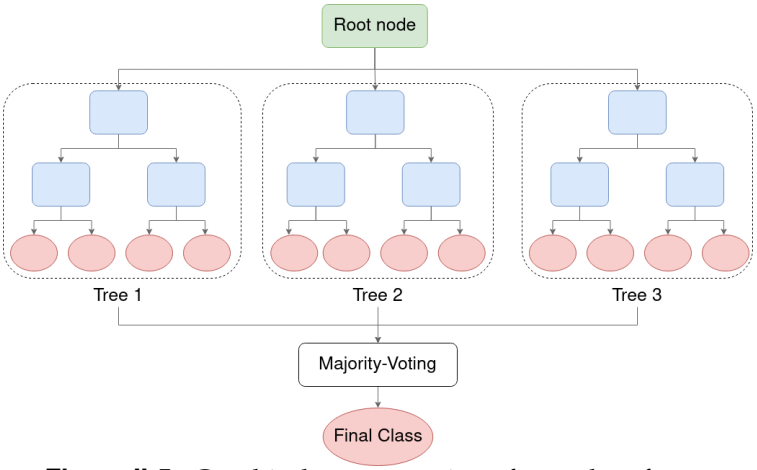



**Figure II-5:** *Graphical representation of a random forest.*

In the literature, many types and supervised ML algorithms can be found, with many variations of the same initial idea, such as support vector machines (Cortes & Vapnik, 1995), Bayesian network (Ben-Gal, 2008), or genetic algorithms (Mitchell, 1998). For further information, the reader is referred to Bishop (2006), which is a mix of mathematical background and ML algorithms), or Mahesh (2020) for a more actual review of ML algorithms.

### II.1.2 Reinforcement learning

Reinforcement Learning (RL) problems consist of learning what to do (how to map situations to actions) to maximize a numerical reward signal. The agent is not told which actions to take, but it must discover which actions yield the most reward by trying them. Notice that actions may affect not only the immediate reward but also the following states and, through that, all subsequent rewards. These two characteristics, trial-and-error search and delayed reward, are the two most important distinguishing features of RL. Therefore, a learning agent must be able to sense the state of the environment, take actions that affect the state, and have a clear goal (or goals) relating to the state of the environment (Sutton & Barto, 1999). This interaction is depicted in Figure II-6.

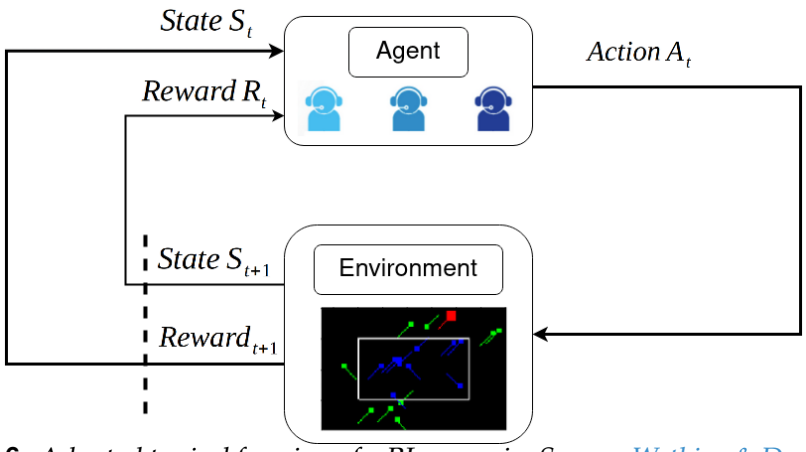



**Figure II-6:** *Adapted typical framing of a RL scenario. Source: Watkins & Dayan (1992)*

One of the challenges that arise in RL is the trade-off between exploration and exploitation. To obtain as much reward as possible, a RL agent must prefer actions that it has tried in the past and found to be effective. However, to discover them, it has to try actions it has not selected previously. The agent has to exploit what it has already experienced to maximize reward, but also it has to explore to make better action selections in the future. Beyond the environment and the agent, we can identify four main sub-elements:

- **Policy**: roughly speaking, it maps the states of the environment to actions (*i.e.*, the strategy).
- **Reward signal**: it defines the goal of the RL problem. It defines what is "good" in an immediate sense.
- **Value function**: specifies what is "good" in the long run. Roughly speaking, it is the total reward an agent can expect to accumulate over the future, starting from a particular state.
- **Model**: it mimics the behavior of the environment. It allows inference to be made about how the environment will behave.

### II.1.3 Single-Agent Reinforcement Learning

A RL problem for a single agent interacting with an environment can be formalized as a finite Markov Decision Process (MDP) described by the tuple $(\mathcal{S}, \mathcal{A}, P, R)$, where $\mathcal{S}$ is the set of states of the environment, $\mathcal{A}$ is the set of actions the agent can take, $P$ is the transition function, being $P(s'|s,a)$ the probability of transitioning to $s' \in \mathcal{S}$, by applying $a \in \mathcal{A}$ in $s \in \mathcal{S}$, and $R$ is the reward function. Notice, in a finite MDP, the sets of states, actions, and rewards ($\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$) have a finite number of elements.

At each time step, the reward is a scalar value, $R_t \in \mathbb{R}$. However, the agent aims to maximize its cumulative reward $G$. That is, maximize both immediate reward and cumulative reward in the long run. Thus, the rewards we set up must truly indicate what we want to accomplish. The cumulative reward, also referred to as return, can be defined as follows:

$$G = \sum_{t=0}^{\infty} \gamma^t R_t, \tag{II.1}$$

where $\gamma$ is a parameter, $0\gamma1$, called the discount rate. It determines how much the agent cares about immediate rewards relative to distant ones.

The RL system aims to find the optimal policy $\pi_*$, which maximizes the expected commutative reward. Let us define the value function of a state $v_\pi(s)$, for a policy $\pi$ (which may not be optimal), as the expected return when starting in the state $s$ following policy $\pi$. For MDPs, we can define $v_\pi(s)$, formally by:

$$v_\pi(s) = \mathbb{E}[G_t|S_t = s]. \tag{II.2}$$

Similarly, we define the value of taking action $a$ in state $s$ under a policy $\pi$ as $q_\pi(s,a)$, providing the expected return:

$$q_\pi(s) = \mathbb{E}[G_t|S_t = s, A_t = a]. \tag{II.3}$$

At least one policy is always better than or equal to all other policies. The optimal policy. Although there may be more than one, all the optimal policies are denoted by $\pi_*$. They share the same state-value function, called the optimal state-value function, denoted $v_*$ and defined as:

$$v_*(s) = \max_\pi v_\pi(s); \quad s \in \mathcal{S}. \tag{II.4}$$

Optimal policies also share the same optimal action-value function, denoted by $q_*$:

$$q_*(s) = \max_\pi q_\pi(s,a); \quad s \in \mathcal{S} \quad \text{and} \quad a \in \mathcal{A} \tag{II.5}$$

Therefore, the optimal policy $\pi_*$ selects what action maximizes the expected commutative reward. If the optimal action-value function $q_*(s,a)$ is known, the best action in the state $s$ is:

$$\pi_*(s) = arg\max_{a \in \mathcal{A}} q_*(s,a). \tag{II.6}$$

The two main approaches used to obtain the optimal policy are policy iteration which manipulates the policy directly, and value iteration, which aims to find an optimal value function adopting a greedy policy (Sutton & Barto, 2018).

### II.1.4 Multi-Agent Reinforcement Learning

A Multi-Agent Reinforcement Learning (MARL) system involves a set of $\mathcal{N}$ interacting agents, which can be cooperative, competitive, or both. It can be described by the tuple:

$$(\mathcal{N}, \mathcal{S}, \{A_i\}_{i\in\mathcal{N}}, \{O_i\}_{i\in\mathcal{N}}, P, \{R_i\}_{i\in\mathcal{N}}). \tag{II.7}$$

At every time step, each agent $i \in \mathcal{N}$ observes a partial representation of the environment $o_i \in \mathcal{O}_i$, and performs an action $a_i \in \mathcal{A}_i$ determined by a policy function $\pi_i$. Then, when an action is taken, the environment evolves to a new state $s' \in \mathcal{S}$ according to the transition function $P$. This transition function depends on the current state and the joint action of all agents. Finally, the reward that each agent receives is given by the reward function. For instance, agents typically share the reward in a cooperative RL; otherwise, they become selfish.

One possible approach for MARL is to train independent agents. However, this simple approach does not perform well in practice (Tan, 1993). To overcome these limitations, in Lowe *et al.* (2017) and Foerster *et al.* (2018), each agent has its centralized critic, only used during learning, that approximates and learns the action-value function given the observations and actions of all agents. However, the critics require the actions and observations of all agents as input. Consequently, their complexity is proportional to the number of agents.

A different solution is proposed in Sunehag *et al.* (2017) to mitigate this scalability issue. In this case, the agents learn an individual action-value function based on their local observations, and the sum of these functions approximates the centralized joint action-value function.

### II.1.5 Q-Learning

Q-Learning (Watkins & Dayan, 1992) is one of the most well-known algorithms based on value iterations. It makes use of a Q-table, which, typically, has the shape [states, actions], and each Q-value $Q(s, a)$ represents the quality of taking as action $a \in \mathcal{A}$, in $s \in \mathcal{S}$. Thus, the standard Q-Learning was designed to work with discrete actions and states.

At each time step $t$, the agent observes the current state $s_t$ and chooses the action $a_t$ with the highest Q-value in that state. After applying the selected action, the agent receives a reward $r_t$, enters on new state $s_{t+1}$, and the Q-value is updated using equation II.8:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big(r_t + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)\Big), \tag{II.8}$$

where $r_t$ is the reward received when moving from state $s_t$ to $s_{t+1}$, $\alpha \in (0, 1)$ is the learning rate, and $\gamma \in [0, 1]$ is the discount factor.

According to equation II.8, the agent adopts a greedy strategy by constantly selecting the actions with the largest Q-value. In that case, it exists the risk of adopting a sub-optimal solution by converging to a local minimum. The $\epsilon$-greedy strategy is widely used to properly explore the state-space, where $\epsilon$ corresponds to the probability of choosing a random action. Typically, $\epsilon$ is initialized to 1 to force high exploration at the beginning, with a decay rate over time to ensure exploitation at the end of the training.

One limitation of this well-known algorithm is the rapid growth of dimensionality in the state-space. The traditional solution is Deep Q-learning (Arulkumaran *et al.*, 2017), which uses a Neural Network (NN) to approximate the Q-values. However, instead of training the NN with the sequence of experiences as they occur during the simulations, they are saved in what is usually called the experience replay buffer. Using a buffer prevents the agent from forgetting past experiences as time evolves and breaks the correlation between consecutive experiences. Finally,

a target network is used to stabilize the learning. The target network is the result of periodically replacing its weights with the ones from the online network used to select the action greedily.

### II.1.6 Deterministic Policy Gradient

Deterministic Policy Gradient (DPG) (Silver *et al.*, 2014) is an actor-critic RL algorithm used for continuous actions that learn a deterministic policy function and a value function simultaneously from an exploratory behavior.

It is not possible to straightforwardly apply Q-Learning to continuous actions spaces because finding the greedy policy would require optimization of $a_t$ at every time step, which is too slow to be practical with large, unconstrained functions approximators, and nontrivial action spaces (Silver *et al.*, 2014). The DPG algorithm uses an actor as the current policy to map states to a specific action. The critic determines the expected reward for an agent starting at a given state and acting according to the previous policy.

As with Q-Learning, it is required to introduce non-linear function approximators to learn and generalize on large-scale state spaces, which means that convergence is no longer guaranteed. However, such approximators appear essential in those scenarios. Lillicrap *et al.* (2015) presented a modification to DPG from Hafner & Riedmiller (2011), inspired by the success of Deep Q-Learning (DQN), allowing the use of NN function approximators. This implementation is called Deep Deterministic Policy Gradient (DDPG), and it was proved that the algorithm could learn policies "end-to-end" directly from raw pixel inputs. Target networks are used to add stability to the training, and an experience replay buffer is used to learn from experiences accumulated during the training.

## II.2 Architecture and approach

The first objective of this thesis is to develop a prototype for the acquisition and preparation of historical data to provide support on the DCB process with predictive and advice capabilities for relevant stakeholders. Concretely, the goal is the development of an infrastructure able to provide data that is used to enhance the prediction of ATFM regulations and study possible downstream effects, focusing on providing advice on the corresponding decision-making process and their resolution.

The prediction horizon is set to the day prior to operations (D-1), where there are no actual ATFM information because most regulations are implemented between 12 and 3 hours before departure. Furthermore, this work aims to provide advice on many relevant stakeholders as possible; thus, the following target end users are identified:

- **Network Manager**, which aims to identify ATFM regulations at the en-route Traffic Volume (TV)[2] level and decide on the extra ground delay;
- **Airlines**, whose flights are directly affected by the regulations due to the associated operational constraints or the issued extra ground delay.

[2]A TV is an environment data structure associated with only one reference location based on geographical entities (e.g., sector, collapsed sectors, or airports). They are used to compare the traffic load and the available capacity (Niarchakou, 2022).

The architecture proposed in this work is organized in three layers as depicted in Figure II-7:

- **Data infrastructure** to support storage and management of data sources required to train the models and provide advice;
- **Predictive capabilities** which comprises the definition, training, and validation of individual machine learning models;
- **Advice capabilities** that uses the trained individual machine learning models to present the information to the end user in a comprehensive manner.

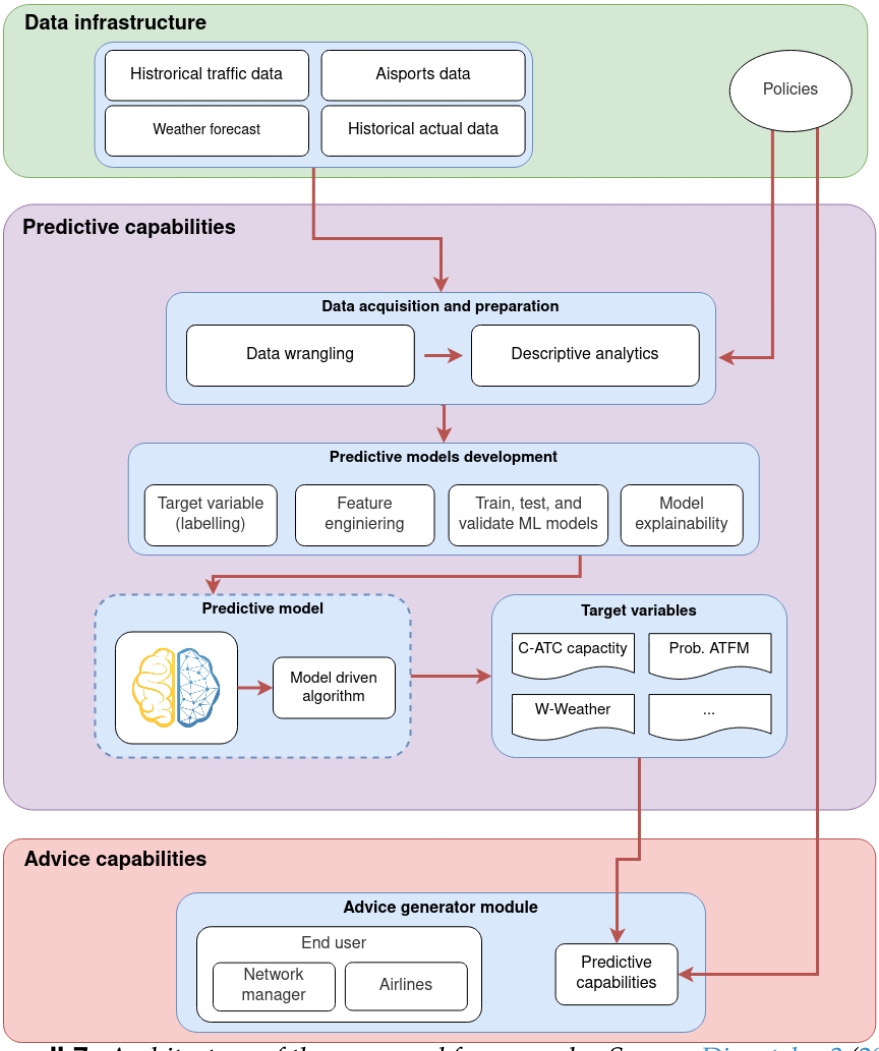



**Figure II-7:** *Architecture of the proposed frameworks. Source: Dispatcher3 (2022)*

### II.2.1 Data infrastructure

An iterative process is used to identify and acquire datasets required by the different machine learning models and development needs. These data are stored and managed in two data infrastructures set up according to the different needs. Small datasets are used locally, while the large ones are stored in Amazon Web Services (AWS). See Section II.3 for more details on the technological solutions and infrastructure characteristics used. The different data sources used by the machine learning models are described in Section II.4.

### II.2.2 Predictive capabilities

The predictive capabilities are developed in two different phases:

1. **Data acquisition and preparations** composed of two activities:
   (a) **Data wrangling** which focuses on the preparations and cleaning of the data;
   (b) **Descriptive analytics** based on data mining techniques to extract the Key Performance Indicatorss (KPIs) used as target variables. It also focuses on identifying precursors for the different target variables.
2. **Models development** consists of:
   (a) **Labelling** because supervised machine learning models work on labeled data, according to the defined KPIs;
   (b) **Feature engineering** is the activity where the selected features are engineered from the raw data and analyzed to see their relevance for the different experiments;
   (c) **Train, test, and validate** of the machine learning models to obtain the system's predictive capabilities.

Different machine learning models have been developed according to different scenarios, goals, and end users. Table II-1 summarizes the main experiments, case studies tackled, and developed ML models:

**Table II-1:** *Experiments, case studies, target users, and ML models*

| Experiment identifier | Case study | Case study description | Targeted user | Model description |
|---|---|---|---|---|
| TV | C-ATC Capacity | Prob. ATFM for TVs | Network Manager | Chapter III |
| | W-Weather | Prob. ATFM for TVs | Network Manager | Chapter IV |
| RL | C-ATC Capacity | Resolution of ATFM | Network Manager | Chapter V |
| Flights | Vueling | ATFM info. for flights | Airline | Chapter VI |

### II.2.3 Advice generator

The outcome of the individual models developed as part of the predictive capabilities has to be integrated to provide meaningful advice to the end user. Moreover, the outcome of the machine learning models might present some discrepancies and uncertainties that need to be considered.

Therefore, the advice generators focus on the following elements according to the previously defined experiments:

- **Selection** of the desired case study and the parameters of the analysis;
- Visualization of the pre-tactical ATFM regulations at the **TV level**;
- Visualization of the possible **resolution** of identified ATFM regulations at the **TV level**;
- Visualization of the ATFM characteristics at the **flight level**.

Finally, the advice generator has a series of requirements that have to be met:

- **Infrastructure:** architecture required to load the data, obtain the predictions from the models, and show the outcome;
- **User-friendly:** the visualization must be easy to understand by the end user;
- **Understandability:** uncertainty in the predictions is always present, so it is required to ensure the user accounts for this uncertainty.

## II.3 Infrastructure and tools

This section details the characteristics of the infrastructures used to develop the different predictive and advice capabilities, which can be summarized as follows:

- **Data lake:** Local, or cloud, storage for the different data sources. Independently of the final size of the data lake used in the different experiments, they have been built to sustain large data-driven projects to train machine learning models;
- **Software and control version**: Programming language, development tools, and control version for collaborative work;
- **External tools**: Additional tools used during the development of the models.

Table II-2 summarizes the resources used in each of the experiments:

**Table II-2:** *Data infrastructures and tools required for each of the experiments*

| Experiment | Case study | Data lake | Computer resources | External tools |
|---|---|---|---|---|
| TV | C-ATC Capacity | Local | Local | R-NEST |
| | W-Weather | Local | Local | R-NEST |
| Flights | Vueling | AWS | Cloud | DataBricks |
| RL | C-ATC Capacity | Local | Local | R-NEST |

### II.3.1 Data lake

A data lake is a centralized repository that allows the storage of structured and unstructured data formats. Compared to hierarchical data warehouses, which store data in specific formats or folders, a data lake is based on a flat architecture and an object storage approach. Some of the advantages of using a data lake in machine learning projects are:

- **Data volume:** Storage is elastic rather than pre-allocated, and the capacity scales with need;
- **Variety:** Data lakes are designed to contain different datasets and formats. Moreover, since all data used are stored in the Data lake, they are always up to date;
- **Centralized:** A centralized storage eliminates problems like data silo or duplication;

As mentioned before, two data lakes have been used during the development of the case studies: local and cloud data lakes. Note that while the overall structure of the lakes is set a priori and unlikely to suffer any modification, with the buckets having a reduced number of major parts, the factual content of the files and folders in those buckets are subject to dynamic changes as the case study advances.

The data in the local data lake are organized to the following structure:

- **input**: Data are stored as received from their source with no modifications;
- **processed**: Data are ready and prepared to be used;
  - **labels**: It contains the different labels to be predicted by the machine learning models, partitioned by the case study;
  - **features**: It contains the engineered and computed featured, partitioned by the case study;
- **samples**: It contains the final samples that will be used to train, validate, and test the models.

The data in the cloud data lake are organized to the following structure:

- **input**: Data are stored as received from its source with none (or minor) modifications;
- **share**: Data are ready and prepared to be used;
  - **sources**: Input data adapted or particularized to the different case studies;
  - **labels**: It contains the different labels to be predicted by the machine learning models, partitioned by the case study;
  - **features**: It contains the engineered and computed featured, partitioned by the case study;
  - **training**: Data sets used to train the machine learning models.
- **samples**: Architecture identically to the /share partition. It contains small samples of each data set.

### II.3.2 Software and control version

Python is the selected language for the development of machine learning models due to the available frameworks, libraries, and community support. However, despite its popularity, it is a high-level language with relatively low computational performance. Therefore, to reduce the computational time required, the feature engineering process is based on parallelization to reduce the required computational time. Similarly, the machine learning models have been trained using libraries that allow the usage of Graphical Processing Units (GPUs). However, notice that this is not mandatory due to the size of the datasets.

Many frameworks are available to train supervised machine learning models and reinforcement learning agents. However, because of their popularity and support, Keras and Scikit-Learn are the frameworks selected to train the supervised models. On the other hand, rather than using the available frameworks for reinforcement learning algorithms, they have been manually implemented to ensure the proper interaction between the algorithms and the environment. On top of the programming language and frameworks, the selected environment to do the development is Anaconda and Jupyter Notebook for prototyping with a document-centric experience.

Last but not least, all the case studies rely on GitHub for software management. This enables collaborative development while keeping track of the code changes and versions. The repositories are structured based on a case-study division.

### II.3.3 External tools

Depending on the nature of the different experiments and case studies, two external tools are required to obtain some of the input features necessary to train the supervised machine learning models, obtain the expected flight plans, or process confidential data. Concretely, it has been required the usage of R-NEST (see Section II.3.3.1) and DataBricks (see Section II.3.3.2).

#### II.3.3.1 R-NEST

R-NEST is a EUROCONTROL model-based simulation tool dedicated to research activities for evaluating advanced ATM concepts. It is a stand-alone desktop application combining dynamic ATFM simulation capabilities with powerful airspace design and capacity planning analysis functionalities (R-NEST, 2022). Although the tool has been designed as an evaluation system, it can also be used to visualize airspace sector configuration and compute complex features. Figure II-8 shows the interface of the tool, where it can be seen the shape of a TV, the expected entry count and workload, and issued ATFM regulations.


**Figure II-8:** *R-NEST interface. ATFM regulation for TV MASB3EH on 6th June 2018*

It is worth mentioning the relevance of R-NEST in the computation of some complex features, such as the workload or the airspace complexity. These features are well-known in the aviation field, and many implementations can be found in the state-of-the-art. However, to the author's best knowledge, there is no official documentation from EUROCONTROL about how to compute them. Therefore, R-NEST is the only option to obtain such input features for the models. The following list contains the most relevant features that can be extracted from R-NEST:

- **Demand/Max demand**: demand counts or max demand counts for Air traffic Control Center (ACC), Air Traffic Control (ATC) sector, or TV;
- **Airport demand/Max demand**: arrival, departure, or throughput counts or equivalent max counts for any airport or group of airports;
- **Occupancy/Max Occupancy**: occupancy counts, or max occupancy counts, for ACC and sectors;
- **Overload / Max overload**: overload values for all sectors and TVs, all for a given peak size (20 min, 30 min, 1 hour, 2 hours) and sliding step;
- **Capacities**: sliding tactical capacity series for any sector or traffic volume for a given integration window and sliding step;

- **Flights list**: flight information list for all flights crossing the selected entity on a given day;
- **Airspace entry list**: flight entry information list for all flight entries of the selected entity on a given day;
- **Delay**: total delay, total daily demand, or total delay per flight expressed in minutes of ATFM delay;
- **Delayed flights**: information list for all flights exceeding a given delay threshold due to regulations of a given reason on the selected entity on a given day.
- **Regulations**: regulation-specific information such as total delay, regulated demand, delay per regulated flight, delayed demand, and delay per delayed flight, as well as regulation reason and sub-period capacities and times.
- **Complexity**: sliding complexity series and summary values for any ACC, sector, and TV for a given integration window and sliding step.

Therefore, because R-NEST is a tool developed by the NM (*i.e.*, EUROCONTROL), it seems to be the perfect tool to compute input features for the machine learning models based on the predictions of ATFM regulations at the TV level. However, one constraint of the tools is the fact that it requires to use of Aeronautical Information Regulation and Controls (AIRACs) (see Section II.4.1.1) as a source of data, which contains snapshots of released historical data.

#### II.3.3.2 Databricks

Databricks (Databricks, 2022) provides a unified, open platform for all the data, teams of data scientists, and developers in a project. It empowers data scientists, data engineers, and data analysts with a simple collaborative environment to run interactive and scheduled data analysis workloads. Databricks builds on the most popular open-source projects, such as Apache Spark, Delta Lake, MLflow, and Koalas, to deliver a true lake house architecture, combining the best of data lakes and data warehouses for a fast, scalable, and reliable data platform.

Built for the cloud, it requires a data lake also allocated in the cloud, which perfectly matches the AWS S3 bucket used for the experiments related to ATFM regulations at the flight level. The use of DataBricks is mandatory due to legal agreements that do not allow the local storage of raw data from Vueling required in one of the experiments conducted in this thesis.

## II.4 Data sources

Flight management activities cover complex tasks that start months before the day of operations and involve different information sources. A wide range of data sources can be found storing information from early stages, such as the flight policies or the Flight Intentions (FIs), to the final post-operational analysis.

When developing machine learning models, ensuring that the data used to train the final models is available during execution time is important. However, one of the challenges in the aviation field is that datasets tend to contain snapshots of released historical data, making it very difficult to know what data were available at a given moment. This is particularly relevant for the network data because planned and released data evolve on time as flight plans are submitted, updated, or canceled. Similarly, proper weather data have to be used according to the prediction horizon of the models. Therefore, for each experiment, it is used the closest possible available information to the prediction horizon.

This section collects and describes all the different data sources used in the realization of the different case studies. Section II.4.1 describes the network data sources (aka. airspace traffic sources). Section II.4.2 presents the numerical weather data sources. Section II.4.3 details the data sets used for labeling. The different data sources, formats, their category, and the available period are summarized in Table II-3.

**Table II-3:** *Summary of the data sources or formats*

| Data Category | Data Source / Format | Case study | Period time |
|---|---|---|---|
| Network data | AIRAC | TV & RL | June, July, Aug, Sept 2018 |
| | DDR2/ALLFT+ | Flights | 2018 |
| | PREDICT | Flights | 2018 |
| Airport characteristics | Airport data | Flights | Static |
| Weather | ERA5 | TV level | June, July, Aug, Sept 2018 |
| | NOAA/GFS | Flights | 2018 |
| | METAR | Flights | 2018 |
| Labelling | EUROCONTROL | TV & RL | June, July, Aug, Sept 2018 |
| | Vueling | Flights | 2018 |

Static data refers to information that does not evolve over time; thus, it remains constant. Examples are the size of the airport or whether the airport is used as a hub by an airline.

### II.4.1 Network data

Network data, as the name indicates, refers to information about the situation of the network. That is, data about the airspace traffic and the characteristics of the network. Typically, traffic information contains a detailed description of the routes according to the planning phase, and the characteristics of the network refer to its configuration. In Europe, the main provider for these data sources is EUROCONTROL, which mainly releases it in two formats: AIRACs and ALLFT+ data. The AIRACs are a detailed description of the flight plans and network configuration (*i.e.*, environment), while ALLFT+ only contains the flight plans information.

The AIRACs are used in Chapter III and Chapter IV as a source of data for R-NEST (see Section II.3.3.1) to compute demand features. For instance, the expected occupancy and entry count, the expected workload, the complexity, the number of conflicts, and the number of flights at different phases. The AIRACs have also been used in Chapter V as a source of pre-tactical flight plans. On the other hand, ALLFT+ data have been used in Chapter VI as a source of information for the FIs and historical pre-tactical flight plans.

Those case studies that use R-NEST to compute complex input features require using the AIRACs. As previously mentioned, R-NEST is mandatory to compute some features due to the lack of documentation. However, the case studies in which all the features are computed from zero will require using ALLFT+ data.

#### II.4.1.1 AIRACs

The AIRACs are a detailed description of the airspace configuration for a period equal to 28 days. It contains information about the network configuration, such as the different ATC sectors, the associated TV, or the opening scheme. It also contains three types of traffic data:

- **Initial (M1) traffic**, which corresponds with the last filed flight plan filled by the airlines excluding ATFM delays;
- **Regulated (M2) traffic** is the same as the last filled flight plan above except that ATFM delayed flights contain a time-offset corresponding to the CASA-calculated ATFM delay, otherwise, is equivalent to the initial trajectory;
- **Actual (M3) traffic** contains the actual trajectories. They start with the regulated trajectory and are updated with radar information when the flight deviates from its more than 5 minutes, 7 Flight Level (FL), or 20 nautical miles.

AIRACs of June, July, August, and September 2019 have been used for the case studies centered at the TV level. That is 112 days (28 days per AIRACs). The traffic information used to compute the different features (scalar variables and artificial images) comes from the *M1* traffic which is the closest traffic information to the prediction horizon.

#### II.4.1.2 ALLFT+

ALLFT+ is the format employed by the historical traffic data provider Demand Data Repository (DDR) [3] to store airspace traffic information. The file format is plain text, which contains information about the flight plans per flight, separated by semicolons. Typically, ALLFT+ files contain 172 data fields that can be classified into six groups:

- **General**: high-level information such as departure airport, destination, departure time, aircraft identification, type of aircraft, o registration mark;
- **Airport Collaborative Decision Making (CDM)**: departure status, collaborative decision-making status, taxi time, and aircraft type;
- **FTFM (M1 – Filed traffic flight model)**: it is the last filed flight plan from the airline;
- **RTFM (M2 – Regulated traffic flight model)**: it only contains information only if the flight has been regulated;
- **CTFM (M3 – Computed traffic flight model)**: 4D trajectory the flight actually followed;
- **Other complementary information**: for example, shortest constrained route (SCR), shortest RAD restrictions applied route (SRR), shortest unconstrained route (SUR), direct route (DCT), and correlated positions report for a flight (CPF);

ALLFT+ data from the entire 2018 have been used for the case studies centered at the flight level. Concretely, because of the desired prediction horizon, *FTFM* traffic has been used as a source of information about the FIs and historical flight plans.

#### II.4.1.3 PREDICT

The PREDICT software is the NM (EUROCONTROL in Europe) support tool intended to estimate the flight plans for the FIs when those still need to be submitted. The software aims to estimate the expected routes mainly using historical data. According to Niarchakou (2022), PREDICT can estimate the flight plans for the next six days (from D-6 to D-1) following the steps below:

[3] DDR is a service provided by Eurocontrol that provides the most accurate picture of pan-European air traffic demand (DDR, 2022)

1. **Enrichment**: The FIs and expected off-block time are compared with the available historical information. Those flights flown in the past (between 6 and 28 days) with the intention to be flown in the future are categorized as confirmed, while the FIs that do not appear in the historical data are considered new flights.
2. **Flight plan assignment**: For confirmed flights, it is assumed that the flight plans will be the same as the available historical data. For the new flights, the route assignment process follows this sequence:
   (a) The software checks the available historical flight plans for the same origin-destination pair in the previous 28 days. If more than one flight plan is available, the searching process uses additional information such as the operator, the day of the week, or the aircraft ID to filter the available routes;
   (b) If no routes are available in step (a), the flight plan is searched in the NM catalog;
   (c) If no available flight plan, it is used the shortest route between the origin and destination airports.
3. **North Atlantic Traffic (NAT)**: Flight plans for NAT traffic is substituted by predictions that consider weather conditions. In these cases, the historically selected flight plans come from days with similar meteorological conditions from the previous three days;
4. **Update**: The predicted flight plans are updated in the DDR portal.

The case study presented in Chapter VI uses PREDICT data assuming that the FIs and Scheduled Off-Block Time (SOBT) are known, but not the flight plans. The required flight plans to estimate congestion features are obtained using a variation of the previous steps. Rigorous implementation of the PREDICT tool is not feasible due to, for instance, the missing access to the NM catalog. However, an implementation purely based on historical routes has been used following the steps below:

1. **Historical data previous six days**: The software searches historical data for the same origin-destination pair in the previous week. If there is more than one flight plan available, the available routes are filtered based on the following information:
   (a) **Day of the week**: Typically, the traffic slightly changes from Friday to Sunday due to the higher demand;
   (b) **Operator**: To take into account possible airline policies or preferences;
   (c) **Aircraft ID**: Final filtering in case of highly frequent origin-destination pairs.
2. **Historical data previous 28 days**: If no available historical information, the analysis is extended using data from the previous 28 days. This second step is used to reduce the computational time required to estimate all the FIs of flight crossing Europe.

PREDICT data from 2018 have been used for the case studies that focus on predicting ATFM regulations at the flight level. Notice that PREDICT is not able to predict flight plans for the first days of the year due to a lack of historical data.

### II.4.2 Numerical weather data

This section describes the numerical weather information used in the different case studies. Two different types of weather data can be found: Numerical Weather Predictions (NWPs) and actual weather information. The NWPs are a collection of processes to predict future weather atmospheric conditions by solving dynamic and physics equations to explain the movements and

changes in the atmospheric conditions. In most of the NWPs, the atmosphere is assumed to be composed of regions that define a set of grid points. The number of grid points defines the resolution of the simulation, and future states for each of them are provided as a prediction. On the other hand, actual weather data refers to recorded atmospheric conditions in real-time.

Realistic atmospheric data, as a function of the altitude and geographical location (latitude and longitude) for different time horizons, can be obtained from many weather forecasts and analyses generated by NWP models. However, in this thesis, European Centre for Medium-Range Weather Forecasts (ECMRWF) and National Oceanic and Atmospherics Administration (NOAA) data have been selected due to their outstanding accuracy and resolution (see Buizza *et al.* (2005) for further details). As a source of actual weather information, it has been used METeorological Aerodrome Reports (METARs) because it is the most common format of observational weather data highly standardized through the International Civil Aviation Organization (ICAO).

#### II.4.2.1 ECMRWF – ERA5

The ECMRWF (ECMRWF, 2022) is an independent intergovernmental organization supported by most of the nations of Europe. Although ECMRWF provides a wide variety of data sources, it has been selected *ERA5* because most of the verification measures indicate that this ensemble forecast has the best overall performance (Buizza *et al.*, 2005). ERA5 is based on the Integrated Forecasting System (IFS), which provides hourly estimations for a large number of atmospheric, land, and oceanic climate variables. The data covers the global atmosphere on a 30 km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80 km. In addition, ERA5 includes information about uncertainties for all variables at reduced spatial and temporal resolutions. Generally, the data are available at a sub-daily and monthly frequency and consist of analyses and short (18 hours) forecasts, initialized twice daily from analyses at 06 and 18 UTC (Hersbach *et al.*, 2020).

Table II-4 summarizes the weather information that can be extracted from this NWP forecast. Notice that because this data source is used to predict en-route ATFM regulations at the TV level, it is assumed a FL equal to 300, which is a frequent cruise altitude (*i.e.*, 10K meters). Furthermore, through the available Python Application Programming Interface (API), only data for the regions of interest have been downloaded.

**Table II-4:** *ECMRWF – ERA5 most relevant weather-related features*

| Name | Description | Units |
|---|---|---|
| Divergence | Rate air spreading out horizontally from a point | $s^{-1}$ |
| Geopotential | Gravitational potential energy of a unit mass | $m^2s^2$ |
| Vorticity | Capacity for air to rotate in the atmosphere | $Km^2kg^{-1}s^{-1}$ |
| Cloud ice water content | Mass of cloud ice particles | $kgkg^{-1}$ |
| Cloud liquid water content | Mass of cloud liquid water droplets | $kgkg^{-1}$ |
| Humidity | Water vapour per kilogram of moist air | $kgkg^{-1}$ |
| Snow water content | The mass of snow (aggregated ice crystals) | $kgkg^{-1}$ |
| U-component wind | Eastward component of the wind | $ms^{-1}$ |
| V-component wind | Northward component of the wind | $ms^{-1}$ |
| Cloud cover | Grid box covered by cloud (liquid or ice) | Dimensionless |
| Ozone mass ratio | Mass of ozone per kilogram of air | $kgkg^{-1}$ |
| Temperature | Temperature in the atmosphere | $K$ |

#### II.4.2.2 NOAA – Global Forecast System

The Global Forecast System (GFS) (NOAA, 2022) is a NWP developed and maintained by NOAA that generates data for dozens of atmospheric and land-soil variables, including temperatures, winds, precipitation, soil moisture, and atmospheric ozone concentration. The system couples four separate models (atmosphere, ocean model, land/soil model, and sea ice) that work together to depict weather conditions accurately.

Specifically, NOAA covers the entire globe with a horizontal resolution of 28 km and look-ahead forecast times up to 192 hours. Data are distributed in GRIB (GRIdded Binary or General Regularly-distributed Information in Binary form) format that allows compression of weather data and includes metadata about the file's content. Although NOAA presents more than 100 different physical parameters, Table II-5 presents the most relevant for weather-related features used in the different case studies. Notice that NOAA data source has been used to obtain weather information at the departure and arrival airports; thus, it is used information from the first available vertical level.

**Table II-5:** *NOAA most relevant weather-related features*

| Name | Description | Units |
|---|---|---|
| Visibility | Horizontal opacity of the atmosphere | $m$ |
| U-component wind | Eastward component of the wind | $ms^{-1}$ |
| V-component wind | Northward component of the wind | $ms^{-1}$ |
| Wind | Nominal wind speed | $s^{-1}$ |
| Ventilation rare | Height multiplied by the transport wind speed | $mph$ |
| Temperature | Temperature in the atmosphere | $K$ |
| Humidity | Water vapour per kilogram of moist air | $kgkg^{-1}$ |
| Ozone mass ratio | Mass of ozone per kilogram of air | $kgkg^{-1}$ |
| Vorticity | Capacity for air to rotate in the atmosphere | $Km^2kg^{-1}s^{-1}$ |
| Cloud mixing ratio | Amount of water vapor that is in the air | $kg^{-1}$ |
| Isobaric_surface | The pressure for each isobaric level | Pa |
| Vertocity | Speed of air motion in the upward or downward direction | 1/s |
| Geopotential | Height of a pressure surface above mean sea-level | $m^2s^2$ |

### II.4.3 Labelling – ATFM information

In supervised machine learning, it is crucial to have a valid ground truth to catalog the different input observations according to the problems of interest. However, it also plays a major role in using reinforcement learning techniques because the agents learn to solve already identified DCB imbalances.

This thesis uses two data sources to label the input observations required to train the models. The samples are primarily labeled depending on whether they belong to ATFM regulations. Nevertheless, if necessary, they are used to catalog observations according to ATFM regulation type, whether the imposed delay was zero, and the location of the regulation.

Supervised machine learning models learn patterns from historical data that are used to provide future advice, meaning that the models will learn from past scenarios and decisions. Therefore, the labeling process of the input observations to predict ATFM regulations at the TV level

has been done using information from the NM (EUROCONTROL). On the other hand, labeling the observations used to predict ATFM regulations at the flight level has been done using information accessible by the airlines. In theory, there should not be differences between data sources. However, the previous approach is suggested to avoid possible downstream effects or to guarantee the possible industrialization of the models. Table II-6 summarizes the labeling sources per experiment.

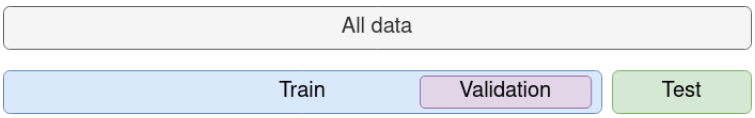
**Table II-6:** *Labelling source per experiment and case study*

| Experiment | Case study | Labelling source |
|---|---|---|
| TV | C-ATC Capacity | Network Manager |
| | W-Weather | Network Manager |
| RL | C-ATC Capacity | Network Manager |
| Flights | Vueling | Airline |

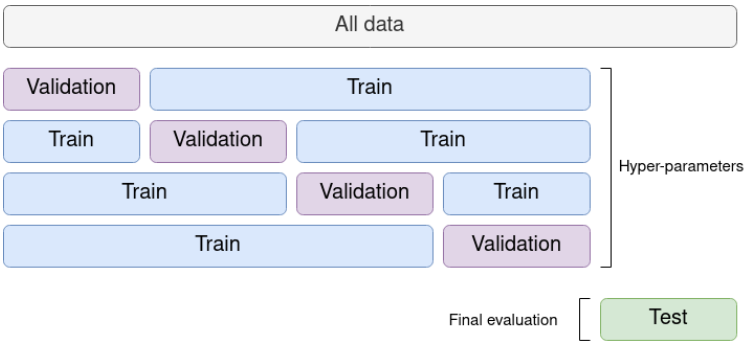
## II.5 Performance evaluation

Once the appropriate machine learning algorithm has been selected, we move into the training and testing of the models. As expected, learning the parameters of a model and testing it on the same data set would cause overfitting. In other words, the model would repeat the labels of the samples it has just seen, obtaining a perfect score in development but failing to predict yet-unseen observations. Therefore, it is common practice to use only part of the data as a training set and hold out part of the available data as a test set, only using it after the models have been trained. The most common techniques are:

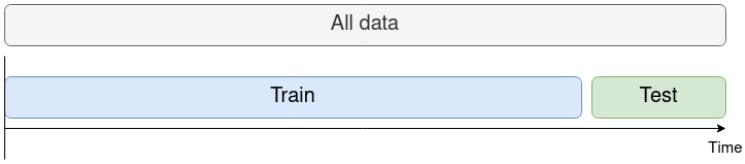
- **Train/Test split** partitions the original data set into two sub-datasets of different sizes: training and accuracy estimation. The typical sizes of the datasets are 80%/20% or 70%/30%. Additionally, the training dataset is usually split into two sub-datasets of different sizes: training and validation. The validation set is used for tuning the model's hyper-parameters.




- **k-fold cross-validation** randomly partitions the training data set into $K$ equal-sized sub-datasets. *K-1* sub-datasets are used to train the models and one for the validation. The process is repeated $K$ times. The final performance of the models is obtained from the test set (Stone, 1978). Variations of this approach are:
  - Leave-p-out cross-validation, which involves using $p$ observations as the validation set;
  - Leave-one-out cross-validation, uses *one* observation as the validation set;
  - Nested cross-validation requires two cross-validation loops;




- **Temporal split** is the sequential split of the data, using the first portion of the time series to train the models and the remaining portion for testing. If desired, the training dataset can be partitioned into the two sub-datasets: train and validation (Dietterich, 2002);




Notice that other variations and combinations of the previous methods can be found in the literature. For instance, the cross-validation with temporal split (Roberts *et al.*, 2017).

In this thesis, the *Temporal split* have been used for small datasets, *i.e.*, 4-5 months, to simulate the most restrictive and realistic possible deployment of the models because ML models are very dependent on data. On the other hand, the random split is selected for datasets covering an entire year to guarantee an equivalent distribution of samples, *i.e.*, avoiding that the models are trained with data until the summer but expected to perform on Christmas when the overload of the network could be different. Moreover, this was a cooperative decision during the development of Dispatcher3 (Dispatcher3 Consortium, 2020). However, the author of this thesis recommends the usage of temporal split for large datasets if it is possible to avoid seasonality issues. An example could be training with data from 2018 and testing them with data from 2019.

### II.5.1 Evaluation metrics

It is paramount to consider the nature, or intended goal, of each of the models when evaluating the performance of the models. In this thesis, four different types of ML models are used, requiring different evaluation techniques according to the end purpose:

- **Classification models** predict a specific class for the input observation (e.g., cat or dog);
- **Regression models** are designed to predict numerical values (e.g., temperature);
  - **Probability distribution** designed to predict the probability distribution of the target variable according to the uncertainty of the models. The approach selected is based on combining regression and classification models (De Falco & Delgado, 2021);
- **Reinforcement learning** where agents learn to interact with a particular environment.

Section II.5.1.1 presents the evaluation metrics for the different classification problems. Section II.5.1.2 shows the selected metrics to evaluate regression models. Section II.5.1.3 exhibits the developed approach to quantify the accuracy and uncertainty when predicting a probability distribution. Section II.5.1.4 summarizes the conventional approach followed to evaluate the performance of the agents when using RL techniques.

#### II.5.1.1 Classification models

There are three main possible approaches to evaluate the performance of ML classifiers:

- **Threshold metrics** which compare the predictions from the models and the ground truth. The prediction is obtained by comparing the probability of the class against a pre-defined threshold (*e.g.*, prob. > 0.5, prediction equal to 1). A well-known threshold metric is accuracy;
- **Rank metrics** are typically used to measure the ability of a classifier to distinguish between classes. For instance, the Area Under the Receiver Operating Characteristic Curve (AUC ROC);
- **Probabilistic metrics** are based on the difference between two probability distributions for a given random variable or set of events. For instance, the Brie Score.

Between the different evaluation metrics for these types of models, the metrics based on thresholds are selected for three main reasons. First, the training of the models has been done using balanced datasets. Second, it facilitates the comparison of the results across experiments due to the number of models. Thirds, they are the most used metrics in the state-of-the-art, which also could facilitate the comparison of results. The selected evaluation metrics are:

- **Accuracy**: Ratio of correct predictions (both positives and negatives);

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- **Recall**: Ratio of actual positives that were correctly predicted;

$$Recall = \frac{TP}{TP+FN}$$

- **Precision**: Ratio of correct positive predictions;

$$Precision = \frac{TP}{TP+FP}$$

- **F1 score**: Harmonic mean of the precision and recall.

$$F1\ score = 2\frac{Precision*Recall}{Precision+Recall}$$

where True Positive (TP) refers to correct positive predictions, True Negative (TN) refers to correct negative predictions, False Positive (FP) refers to wrong positive predictions, and False Negative (FN) refers to wrong negative predictions. Figure II-9 depicts each possible category per prediction according to the target label.

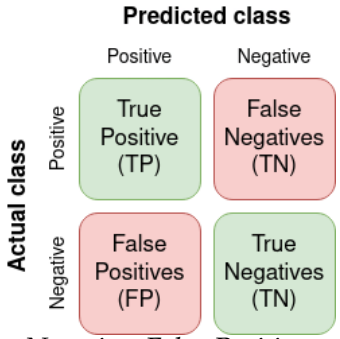



**Figure II-9:** *True-Positive, True-Negative, False-Positive, and False-Negative predictions*

#### II.5.1.2 Regression models

There are many evaluation metrics available to quantify the performance of regression models. For instance, three of the most used ones are:

- **Mean Square Error (MSE) / Root Mean Square Error (RMSE)** is an absolute measure of the goodness of the fit. It is calculated by the sum of the squared errors divided by the number of data points;
- **Mean Absolute Error(MAE)** is similar to MSE. However, MAE takes the sum of the absolute value of error;
- **R square** is a good measure to determine how well the model fits the dependent variables. It is calculated by the sum of squared prediction errors divided by the total sum of the square errors replacing the calculated prediction with the mean.

The MSE gives larger penalization to large prediction errors by squaring them, while MAE treats all errors equally. MAE has been selected as most of the expected delays are going to be close to zero, and it has been decided not to over-penalize mistakes due to large ATFM delays as the reason for them is not recorded; thus, it is unknown in almost all the observations.

#### II.5.1.3 Probability distribution ATFM delay

In this thesis, the prediction of probability distributions is based on the combination of regression and classification models. The goal is not just to provide a real number from a regressor model but to take into account the inherited uncertainty present in the ML models. As the end goal is to estimate the uncertainty in the prediction, the evaluation metrics are designed with the same intention.

The **accuracy** of these models is computed as the difference between the expected value of the distribution and the actual ATFM delay. Therefore, the Mean Absolute Error (MAE) can be computed to quantify how close the expected value is to the actual ATFM delay.

As the classifier is trained to capture a continuous variable in a range of possible values after a binning process, it is possible to define a **measure of uncertainty** considering the range covered by a given distribution percentile (De Falco & Delgado, 2021). The average minutes required to cover 90% of the probability in the distribution is used to measure this uncertainty. The lower the uncertainty, the narrower the distribution; therefore, fewer minutes are required to cover 90% of the probability.

Although the probability distribution provides a range of possible values, there may still be cases where there are significant discrepancies between the predicted distribution and actual ATFM delay. To better understand these extreme cases, *i.e.*, when the actual ATFM delay is much larger or smaller than the values predicted by the distribution, the number of **hits** is calculated. The number of hits represents the percentage of times the actual ATFM delay falls within the predicted probability distribution. Note that the classifier used to characterize the distribution is bounded by the discretization of the error of the regressor as described in Section VI.4.4.

#### II.5.1.4 Reinforcement learning

In RL, the reward function is an incentive mechanism that uses reward and punishment to tell the agent what is correct and what is wrong. It is a real value that shows the agents how good or bad the actions taken were. Therefore, one of the most widely used indicators to evaluate the agents' performance is the sum of rewards earned at the end of each episode. However, to complement the evaluation process, other KPIs related to the system's goal are taken into account. For instance,

KPIs linked to the resolution of ATFM regulations are the overall delay issued to the flights or the number of regulated flights.

Note that the reward received by all agents in the system is taken into account to ensure that the agents behave as expected. In cooperative environments like ATM, it is important that agents share the reward; otherwise, they will become selfish like in competitive environments.

## II.6 Model explainability

Understanding the reason behind the predictions done by complex AI systems is crucial to ensure compliance with company policies, industry standards, and government regulations (Force & Daedalean, 2020). Moreover, it shows stakeholders the value and accuracy of the findings. Typically, they are considered "black boxes", and it can be difficult to disentangle how the model arrives at a certain conclusion. Therefore, interpreting and understanding the reasons behind the predictions done by the models becomes all the more important in critical scenarios such as ATM.

Model explainability is a key component for a solid human-machine interface, allowing partial levels of automation and human users to interact with powered AI systems in a meaningful and collaborative way. With that goal, two analyses are performed: first, a *confidence-level analysis*; second, a game theory approach called *SHapley Additive exPlanations (SHAP)* (Lundberg & Lee, 2017). The *confidence-level analysis* aims to show how sure the model is about the predictions it makes and the trend of the models. The larger the probability, *i.e.*, closer to one, the more confident the model is about the regulation. *SHAP* is used to try to explain the output of the models, assigning contribution scores by optionally giving separate consideration to positive and negative contributions; therefore, identifying which input features are more relevant for the trained model.

### II.6.1 Confidence-Level analysis

The proposed confidence-level analysis aims to study the probabilistic output of the models based on using MLPs. The analysis shows the activation values of the neurons in the output layer, *i.e.*, the probability, according to the prediction of the models and the ground truth. It analyzes the true-negative, true-positive, false-negative, and false-positive predictions with respect to the outcome of the models.

The end goal is to visually analyze the confidence of the system when predicting ATFM regulations, ensuring that the models have been appropriately trained. If the confidence in the prediction is high indicates the patterns learned from the historical data are stable.

### II.6.2 SHAP analysis

In many real-life applications, especially those with high safety levels, the performance of the models is as important as its interpretability. That is, obtaining theoretical guarantees on the expected behavior of machine learning-based systems during operation. This section assumes the models have been trained, and now the goal is to understand the predictions, *i.e.*, trust the outcome of the models and gain insight into the factors impacting them.

To understand the factors impacting the predictions, SHAP is used as it is able to explain the output of any machine learning model (Lundberg & Lee, 2017). This technique is widely used in ML applied to ATM; for instance, Mas-Pujol *et al.* (2022) employed SHAP to study the influence of both scalar and image-based input features predicting the likelihood of traffic volumes to be regulated. Dalmau (2022) used it to understand the outcome of the proposed ensemble method to

predict the likelihood of re-routing to mitigate ATFM regulations. Lambelho *et al.* (2020) utilized SHAP to explain the models used for strategic slot flight assignment at London Heathrow Airport. Xie *et al.* (2021) used it in a more general manner to explain ML solutions in ATM.

SHAP calculates the contribution of each input feature (or pixel when using images) to each prediction made by the model, which can be positive or negative, based on the concept of Shapley values from cooperative game theory. Luo *et al.* (2021) found that SHAP provides a more intuitive and interpretative way of understanding the relationships between input features and the model's predictions than other model-agnostic techniques such as Local Interpretable Model-Agnostic Explanations (LIME).

## II.7 Advice capabilities

The advice capabilities of a ML system are as important as its performance. It is crucial to present the right information at the right level of detail to ensure meaningful advice. Furthermore, this module is linked to the end-user policies because, with the same set of predictions, different advice can be provided according to the end-user preferences. Therefore, the goal of the advice generator module is to collect all the information from the machine learning modules, including possible information about the quality of the predictions, and build a support tool to help the decision-making process by providing advice to stakeholders based on the subset of predicted KPIs.

The framework shall focus on readability and interpretability, avoiding information overflow (Edmunds & Morris, 2000). Visual representation of the information is paramount to ensure that the end-user understands both the predictive analytics provided and the probabilistic nature of the information. Therefore, the advice generator has to consider the interpretability accuracy trade-off. To this end, the proposed advice generators will be composed of the following axes:

- **Requirements**: The advice generator needs to provide a meaningful, readable, and easy-to-interpret visualization of the models' outcome;
- **Architecture**: The main goal of the advice generator is to assemble a set of specialized models. Therefore, a software architecture is required to ensure the right data and models will be used at each moment, enabling the connection between the different models and the generation of the desired visualization. This requires the following components:
  - **Data manager**: Within the advice generator architecture, a data management infrastructure is required to reduce as much as possible the computational cost of gathering the required data;
  - **Controller**: Entity used to ensure that the requested information will be visualized, and therefore, responsible using the appropriate ML models are;
  - **Outcome**: Suitable visualizations of the models' outcome focusing on interpretability.

Different advice capabilities are developed for the different case studies to experiment with different representations, taking into account the possible preference of the end user. For the case studies where the NM is the end user, Section II.7.1 presents a web application emulating the appearance of the current tools used has been developed. Moreover, at the moment of writing the thesis, it is expected to integrate the models into R-NEST thanks to the collaboration with EUROCONTROL (see Section II.7.2). For the case studies where the airspace users (airlines) are the end users, an integrated view of the different models has been created, focusing on only showing relevant information and taking into account the models' inherited uncertainty (See Section II.7.3).

Moreover, a second approach based on visualizing the impact and severity of possible ATFM regulations through the planned rotation of a specific aircraft frame is presented in Section II.7.4. Finally, Section II.7.5 introduces the outcome of the RL system merged with the expected pre-tactical traffic to provide visual advice on what flight could be optimal to delay.

### II.7.1 Web application

The web application developed as an advice generator has been implemented using the framework Django (DJANGO, 2005) and a Model-View-Controller (MVC) architecture (Deacon, 2009). Django is a high-level open-source Python web framework that was designed for fast prototyping, rapid development, and pragmatic design. MVC is a very well-known architecture for developing user interfaces that divide the logic of the program into three interconnected elements:

- **Model** in charge of managing the data of the application;
- **View** for representation and rendering of any information;
- **Controller** responds to user inputs and contains the logic of the application.

Figure II-10 depicts the architecture used to develop a web application as the advice generator. The *Controller* collects the parameters the user selects and creates 30-minute intervals between the selected start and end timestamps. The models are designed to handle samples with that specific time length. Then, the *Model* provides the input features for each of the 30-minute samples and the expected open scheme for the selected day. The input samples are fitted into the ML models, and the *View* uses those predictions and the open scheme to create the final advice. The *View* uses the open scheme to represent when the TV is expected to be operative.

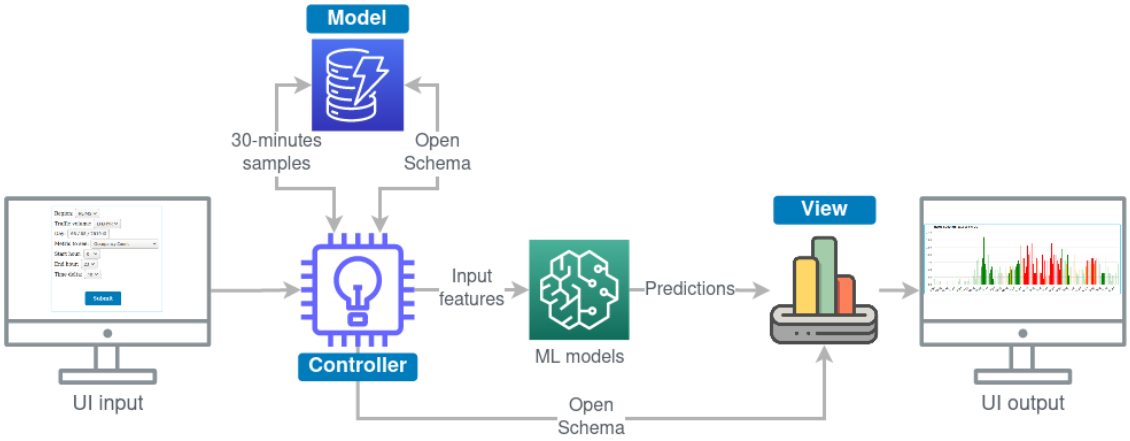



**Figure II-10:** *Advice generator architecture ATFM regulations for the NM (network level)*

It is worst to mention that the developed application contains only some of the required security measures to make it public. All the tests have been done using local or private servers; thus, additional actions will be required to ensure a successful industrialization if desired.

### II.7.2 Integration into R-NEST

As mentioned, R-NEST is a model-based simulation tool developed by EUROCONTROL and used to evaluate advanced ATM concepts. Although the final integration of the models has to be done with the collaboration of EUROCONTROL, this subsection explains and shows the required steps to do such an integration.

The defined road map in conjunction with EUROCONTROL can be summarized as follows; where at the moment of writing this thesis, steps 1, 2, and 3 have been done:

1. **Coordination** with EUROCONTROL to define a plan;
2. **Convert** the ML models from Python to C++;
3. Independent **validation** of the models in C++;
4. **Integrate** the models into R-NEST;
5. **Validate** the integration in R-NEST.

Before starting the integration of the models, it is paramount to have a stable version, ensuring that the required input features can be computed and fitted into the models to obtain the predictions. Next, R-NEST has been developed using C++; thus, the models must be converted to this programming language to avoid possible problems during execution time. Finally, the visualization of the results will be integrated into the current tool in R-NEST used to visualize DCB issues. Figure II-11 shows the expected result combining the current visualization of the prediction of ATFM regulations from the ML models.

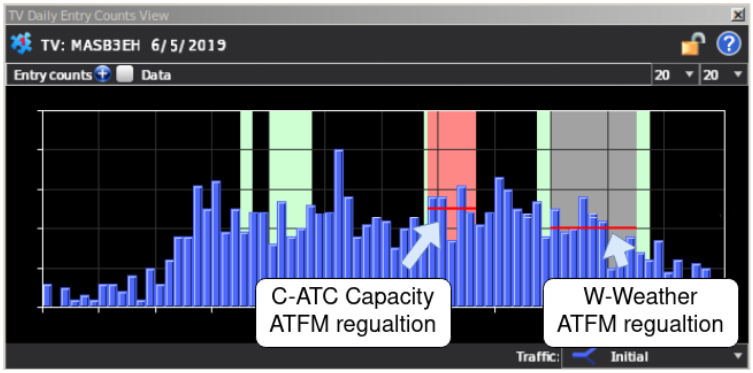



**Figure II-11:** *Example integration of the models into R-NEST*

The initial models were developed using Keras (Keras, 2015), an open-source Python API created to train different types of machine learning models. There are two possible approaches to developing the required models in C++:

1. Develop the ML models **directly using C++**;
2. Convert the trained models into C++ using **third-party libraries**;

The second option is selected to avoid re-building from zero the models. There are plenty of well-known libraries to convert models from Python to C++, but it is crucial to consider that the end user is EUROCONTROL, making necessary the use of a third-party library with a license that allows any possible usage. To this end, the API *frugally-deep* (frugally-deep, 2018) has been selected because it is a specialized API to convert Keras machine learning models to C++.

The selected API converts the Keras model (H5 format) into a JSON, which later is used to replicate the model's architecture, taking into account the value of the weights in each layer. Then, the built-in functionalities allow the user to obtain the prediction from the converted models.

Finally, after converting the models to C++, the performance has to be validated to ensure its correct behavior. No loss of information during the convention can be guaranteed if both models' accuracy, recall, precision, and F1-score are the same.

Lastly, the steps covered in this thesis are the conversion of the models from Python to C++ and the independent validation. For the final integration, precise and detailed documentation about how to compute the required input feature and how to execute the models has been provided to EUROCONTROL. However, further discussion with the team developing R-NEST will be required for the final integration.

### II.7.3 Integration view ATFM for specific flights

Airlines need to monitor flights affected by ATFM delays closely and actively produce new flight plans and solutions to reduce the impact of this delay on their fleet. It is paramount as soon as possible for effective fleet management not only if a flight is impacted by ATFM delay but the characteristics of this (amount of delay and type of regulation). Figure II-12 presents the pipeline of the framework for ATFM regulations at the flight level, where four different ML models are required. First, the advice generator evaluated the likelihood of flight being regulated. Second, for regulated flights, it extends the analysis to provide the expected ATFM protected location and if the ATFM is going to be zero. Finally, the probability distribution of ATFM delay for non-zero regulated flights.

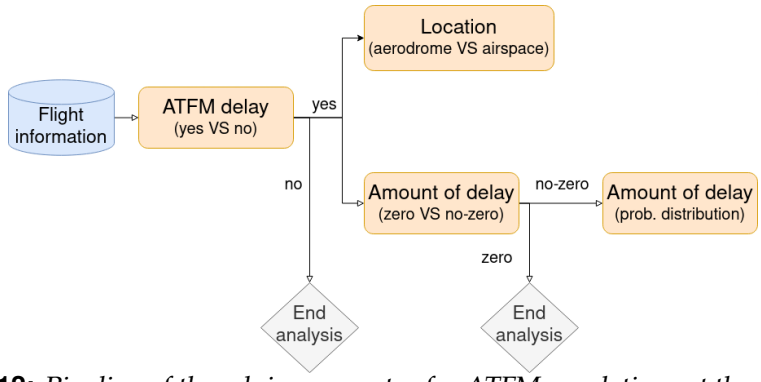



**Figure II-12:** *Pipeline of the advice generator for ATFM regulations at the flight level*

The proposed integration view consists of combining the outcome of the different models and the presented pipeline, ensuring that only the necessary information is displayed, and using a color scheme to indicate the uncertainty of the models. Uncertain predictions are shown in red; otherwise, green is used. Figure II-13 shows, as an example, the final integration view for a flight from *LEBL* to *LWDE* on September 12$^{th}$ 2018 operated by Vueling. The example shows the advice capabilities for a regulated flight with an expected delay different than zero, but the integration view adapts according to the outcome of the models.

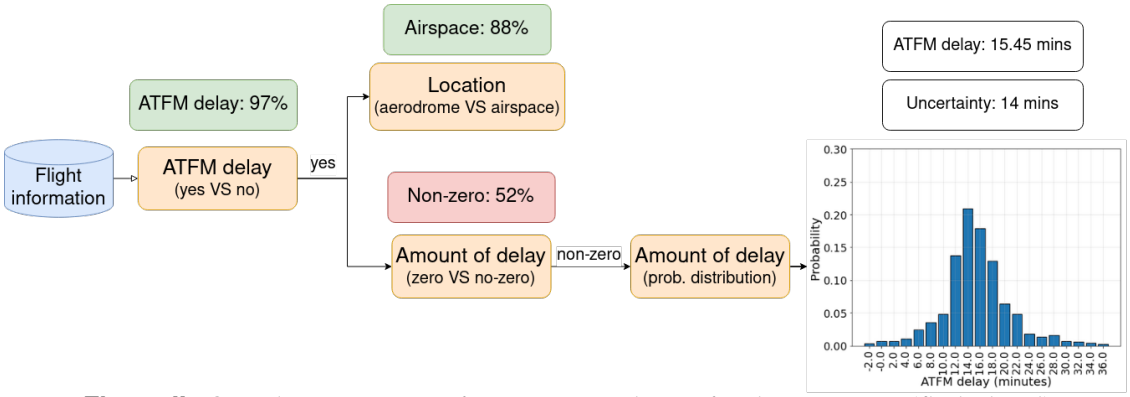



**Figure II-13:** *Advice generator for ATFM regulation for the operators (flight level)*

### II.7.4 Reactionary delay

It is important for airlines to understand the implications of disruptions on current flights on their fleet. This is particularly relevant for the propagation of delay (and cost) due to reactionary delay.

An estimation of how flights will propagate ATFM delays through the network can be achieved by simulating how the delay will propagate by the sequence of planned flights using a do-nothing approach. Instead of relying on classical methods such as Monte Carlo simulation, it is possible to explicitly consider the time distribution of the different processes involved (block time and time on-ground) and combine them (convolution) to obtain a probabilistic representation. FigureII-14 shows the usual division of the different flight phases and the standardized procedures associated with them:

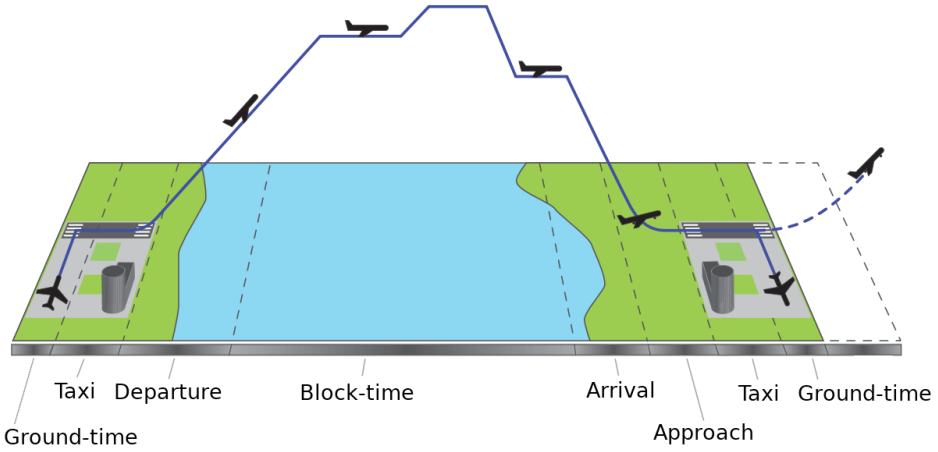



**Figure II-14:** *Different flight phases in a Instrument Flight Rules (IFR) flight. Source: Prats (2011)*

It is important to consider the whole distribution of possible delay rather than the expected delay propagated as this has an implication on the probability of breaching a curfew (only the tail of those distributions) and on the expected cost of delay (as cost of delay grows non-linearly with delay). To this end, the set of ML models developed to provide advice on specific ATFM characteristics at the flight level are combined with other operational parameters to assess the potential impact of disruptions in the network and possible reactionary delays. Concretely, this advice generator is centered on predicting the potential propagation of reactionary delay for Vueling flights with models trained 24 hours prior to SOBT. The architecture of the tool has been inherited from Polit3 (Pilot3, 2022), and it has been adapted to provide advice for D-1, using the ATFM models developed in this thesis. Figure II-15 is a high-level representation of the convolutional process used to propagate possible reactionary delays through multiple rotations.

Examples of possible outcomes of the system are the likelihood of missing a slot or the potential breaching of curfews due to ATFM delays, which are non-observable actions in historical datasets and cannot be predicted using conventional supervised machine learning models.

It is worth mentioning that this approach focuses on possible downstream effects of issued ATFM delays to the flights, which is the topic of this thesis. However, to have a complete view, it will be necessary to consider other possible rotational and non-rotational reactionary delays, possible holdings, and extra airport delays, among others. It will be necessary to create the corresponding ML models or heuristics models to take into account these additional factors.

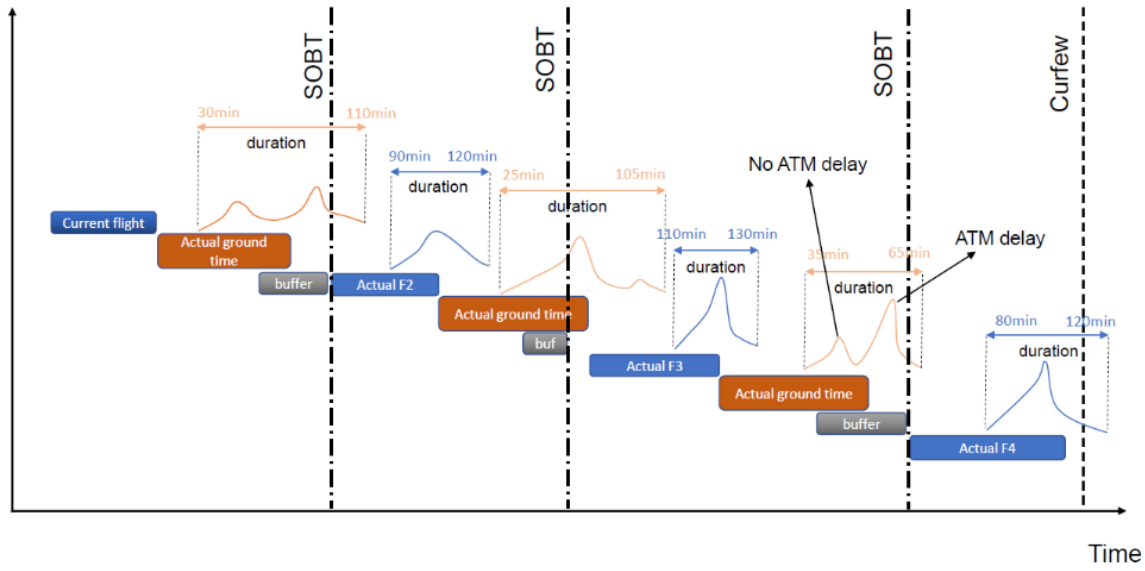



**Figure II-15:** *Convolutional process based on a do-nothing approach for the reactionary delay. Source: Dispatcher3 (2022)*

### II.7.4.1 Architecture

Three main elements are required to estimate the propagation of ATFM delay along the different rotations. First, developing a set of estimators to predict each aircraft operation is necessary. Second, specific data infrastructure is required to obtain the information needed to execute the models. Third, it is necessary to define the modeling approach to predict rotation.

The complex estimators are composed of a set of other individual estimators and a data infrastructure called *estimator data*. The estimators have two functionalities: initialize and estimate. On the other hand, the estimator data has as an attribute a *DataGatherer* responsible for obtaining the required information through the *DataRawGatherer*. This last component is the change of accessing the data lake and collecting the required information to run each estimator.

Figure II-16 presents the class diagram of the system. Each box corresponds to one of the main classes, where the name is presented at the top in bold. Next, the required attributes for each of them. Finally, the methods each class has.

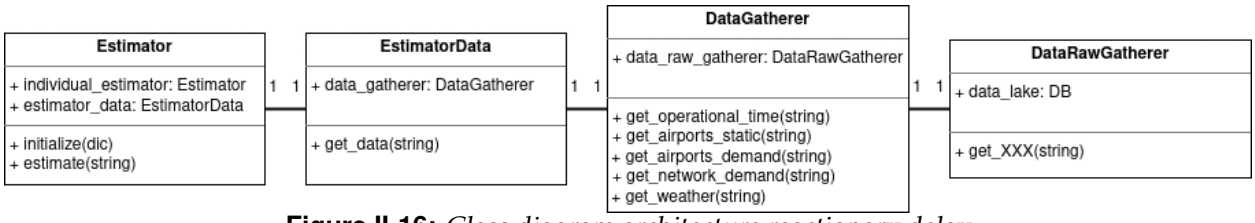



**Figure II-16:** *Class diagram architecture reactionary delay*

### II.7.4.2 Individual estimators

Individual estimators are required to estimate the different phases present in the aircraft operations for the different rotations of an aircraft frame. These estimations are based on computing probability distributions of the required time for each of the phases, which include:

- **Block-time**: gate-to-gate time, including all processes from leaving the gate at the airport of origin to arriving at the gate at the destination);

- **Turnaround processes**: including all processes of de-boarding, re-boarding, cleaning, refueling (if needed), among others;
- **Buffers**: which can be either part of the turnaround process or the arrival time of the flight;
- **Delays**: unexpected delays issued, such as ATFM delays.

Table II-7 provides a more detailed description of the characteristics of the developed individual estimators required to estimate the possible impact, or severity, of issued ATFM regulations for specific flights. As well as the type of estimator required.

**Table II-7:** *Characteristics individual estimators*

| Estimator | Description | Model type |
|---|---|---|
| Block Time | Time from SOBT to SIBT | Heuristic (SIBT-SOBT) |
| Minimum ground time | Minimum turnaround time for a given flight | Heuristic (Based on values from De Falco & Delgado (2021)) |
| Departing within CTOT | If ATFM delay is issued, uncertainty on when the flight will depart within the slot [-5,+10] minutes | Heuristic (triangular distribution centered in zero ranging from -5 to +10 minutes) |
| Probability ATFM delay | Probability of flight being regulated due to ATFM. If the flight is known to be regulated, the probability would be 1. Otherwise as computed by VI.5.2 | Machine learning (see Section VI.5.2) |
| Probability zero minutes ATFM delay | Probability of flight after being regulated having an ATFM delay of zero minutes assigned | Machine learning (see Section VI.5.4) |
| ATFM delay if positive | Amount of ATFM delay if regulated and positive delay assigned | Machine learning (see Section VI.5.5) |

Note that these estimators could provide either a value (e.g., the heuristic estimators are built in that manner) or a distribution if uncertainty is present (e.g., ATFM delay and hence Calculated Take-Off Time (CTOT)). When only a value is produced, this is considered as a probability of certainty of being that value. However, using probability distributions allows us to use the convolution of distributions as an underlying process to add the duration of the different processes involved in the flight rotations.

It is worth mentioning that all the architecture described estimating the reactionary delay could be reused if each of these individual models is substituted by improved versions, e.g., block time could be the combination of taxi-in, taxi-out, and take-off to create landing models as introduced in De Falco & Delgado (2021).

#### II.7.4.3 Modeling approach

The convolutional process estimates the required time for each phase in the aircraft operations along the different rotations. The first step is to estimate when the **aircraft will be ready for the next rotation**, considering the expected departure and arrival time for the current rotation. Second, the departure time of the next rotation is estimated taking into account the **probability of having ATFM delay**. Third, with the previous information and the **expected block time**, it

is possible to estimate the arrival delay of the next rotation. Finally, the possible delay can be propagated through the different rotations following the same steps. Figure II-17 describes the interaction between these processes.

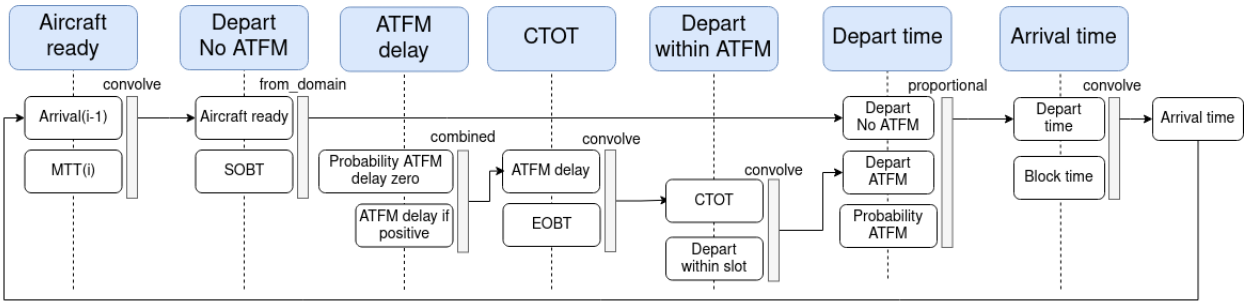



**Figure II-17:** *Block diagram of the process estimate the different flight phases in a IFR flight*

Next, each of the required flight process estimators is detailed. Note that, as previously mentioned, each of these processes is either a time or a distribution of possible times depending on the outcome of the different estimators:

- **Aircraft ready**: convolution between the arrival time of the previous rotation and the minimum turnaround time estimation;
- **Departure time without ATFM**: based on SOBT and aircraft ready time, i.e., flight departing at SOBT if it is ready before, otherwise departing when aircraft is ready;
- **ATFM delay**: if flights are already regulated (or too close in time < 4 hours from current time), the information on their ATFM status is considered fixed as in the fleet status obtained. For the remaining flights, the machine learning models of ATFM delay are used to estimate their probability of being regulated and the amount of delay experienced;
- **CTOT**: available CTOT in flight plan information for a given flight if the time horizon is smaller than three hours. Otherwise, current EOBT and ATFM delay;
- **Departure with ATFM**: Convolution of CTOT and departing within CTOT slot distribution which captures the uncertainty on when the departure will happen within the slot;
- **Departure time**: For the first flight on the sequence of rotations to estimate, it is assumed a departure at EOBT with a probability equal to one. For the other flights, the departure time is computed by combining (as a function of the probability of ATFM delay):
  - **Departure time without ATFM**: based on when aircraft would be ready (aircraft ready) and the available SOBT;
  - **Departure time if the flight is regulated by ATFM**: based on the CTOT estimated and the probability of Departing within the CTOT slot.
- **Arrival time**: convolution between the departure time and the block time

From all the previous individual estimators and the intermediate steps of the convolutional process, it is possible to provide useful advice to the duty manager from the airline who is planning the different rotations of an aircraft frame for a specific day of operations. For instance, it is possible to provide the probability distribution of depart/arrival time, ATFM delay, or the aircraft ready time. Nevertheless, it is also possible to provide numerical advice such as the probability of missing the ATFM slot, the expected buffer per rotation, the probability of breaching a curfew, or the average reactionary delay. Therefore, two levels of granularity are provided when using the

ATFM models to provide advice on reactionary delay. Visualizing the probability distributions can help understand the overall citation better, while numerical advice shows the likelihood of a particular event, such as breaching the curfew.

### II.7.5 Reinforcement learning

The outcome of the RL system is the required ATFM delay (ground delay) required to smooth already identified DCB issues. The system provides the minutes of delay per aircraft fame crossing the TV that should be regulated.

The system tries to solve demand-capacity imbalances using 30-minute intervals. Therefore, the advice generator will provide advice for the same time period. Concretely, the tool uses a simple color scheme to indicate which flights the system is suggesting to regulate:

- **Red**: System-suggested flights for regulation
- **Green**: Non-regulated flights outside the sector in the corresponding timestamp
- **Blue**: Non-regulated flights inside the sector in the corresponding timestamp

The goal of the advice generator is to visually show which of the planned flight should be regulated to smooth the demand-capacity imbalance. A set of images showing the location of the flights at a specific timestamp for a particular TV are displayed using the previous color scheme. The idea behind this is to try to see whether the system is showing any pattern of behavior or if there is a preference when delaying flights. Figure II-18 is an example of the proposed advice capabilities for the RL system for a particular timestamp.

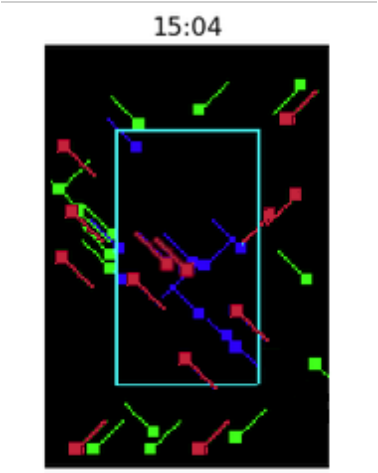



**Figure II-18:** *Advice capability example smoothing ATFM regulations using RL techniques*

*If I only had an hour to chop down a tree, I would spend the first 45 minutes sharpening my axe*

— Abraham Lincoln

# III

# C-ATC Capacity ATFM regulations

As presented previously, sector configuration and flow management solutions (*e.g.*, STAM measures) are the first steps in aligning expected demand and capacity. However, if demand still does not meet capacity, Air Traffic Flow Management (ATFM) measures are implemented to smooth demand. Currently, the most common measure in Europe consists of reducing the rate at which aircraft enter the congested Traffic Volume (TV) for a period of time. The flights subject to one or more regulations are issued with a ground delay by the Computer Assisted Slot Allocation (CASA) system, a simple heuristic algorithm based on first-planned-first-served.

The different Air Navigation Service Providers (ANSPs) across Europe, through the Flow Manager Position (FMP), in collaboration with the Network Manager (NM) operators, are in charge of deciding when and where to apply regulations to solve demand-capacity imbalances. This implies that the global network delay is primarily controlled by the judgments of different humans to solve their local Demand-Capacity Balancing (DCB) problems. Although humans can handle simple scenarios with moderate overload efficiency, it is expected to deteriorate in critical scenarios because of the complexity and interactions between different active regulations.

The introduction of new support tools for the FMPs in the **detection phase** could reduce the amount of work, or at least the difficulty, of their operational tasks. Indeed, it could even result in a capacity increment. This Chapter proposes the use of **supervised Machine Learning (ML) techniques to detect airspace ATFM regulations** during the pre-tactical phase when it is required to identify major DCBs issues. Concretely, this work focuses on two different Neural Networks (NNs) capable of replicating the human decisions made in the past to identify where and when en-route ATFM regulations are necessary. First, a Recurrent Neural Network (RNN) that uses scalar variables. Second, a Convolutional Neural Network (CNN) that uses images. Third, a hybrid architecture combines the previous RNN and CNN.

## III.1 State of the Art

Following pioneer work done by (Odoni, 1987) to improve ATFM performance, the literature shows three main trends: proposals without any Artificial Intelligence (AI), approaches using supervised ML, or works exploring Reinforcement Learning (RL) techniques. All three families are present in the Single European Sky Air traffic management Research (SESAR) program, leading the investigation into the future of ATFM in Europe.

Several studies have dealt with DCB problems in the Air Traffic Control (ATC) network without the use of AI techniques. For instance, Tang *et al.* (2012) identified the gaps in existing 3D sectorization methods and presented a new approach based on minimizing four different Key Performance Indicatorss (KPIs). Similarly, Graña (2019) presented a detailed literature review on computational approaches to improve airspace configuration and solve DCB issues. Other examples are Melgosa *et al.* (2019), where trajectory optimization is used to alleviate DCB problems, and Xu & Prats (2018) that presented a method to introduce linear holding to absorb ATFM. However, these approaches assume that the DCB has been detected, focusing on possible solutions.

Applying ML techniques to Air Traffic Management (ATM) is a very active area of research. It has proved to be successful in applications such as predicting Air Traffic Controllers (ATCOs) workload (Gianazza, 2010; Gianazza & Guittet, 2006), estimating the airspace complexity (Isufaj *et al.*, 2021), trajectory prediction (DART, 2019; Cheng *et al.*, 2021), or predicting the total network delay (Sanaei *et al.*, 2021). Despite the research activity conducted on machine learning applications to ATM in the last years, there is a significant gap in tackling the detection of DCB issues leading to ATFM regulations. In particular, and to the best of the author's knowledge, there is no existing literature on the identification of ATFM regulations using supervised ML models for purely related demand issues at the TV level.

Even though no research has been conducted concretely on the detection of ATFM regulations using AI techniques, this is implicitly done in those approaches based on RL techniques. For instance, Barnhart *et al.* (2012) presented a fairness metric to measure deviation from first-planned-first-served in the presence of conflicts, and more related to the topic of this Chapter, Kravaris *et al.* (2018); Chen *et al.* (2021) presented a multi-agent RL system based on ground delay.

In order to fill the gap in the literature, this work focuses on the detection of ATFM regulations for en-route TVs. It aims to create a support tool that replicates past decisions made by the FMPs to help detect more efficiently and faster possible ATFM regulations due to demand–capacity imbalances. Although not optimal, the models will replicate what they have learned from past actions in future scenarios.

## III.2 Problem formulation

Demand-capacity imbalances leading to ATFM regulations are particularly complex on the European ATC network. FMPs must agree on where and when these regulations are going to be required to smooth an unsafe amount of expected traffic. The ATFM regulations are mainly characterized by seven elements: the date, the regulation ID, the TV associated with the regulation, whether the regulation is for en-route or airport traffic, the start timestamp, the end timestamp, and the regulation reason that best indicates the reason for such regulation.

Different approaches could be used to automatize the detection of ATFM regulations. However, the use of *supervised machine learning* techniques aims to create a system that learns from past scenarios and replicates patterns from historical data. Thus, it is a system that will replicate the decision taken in the past to future scenarios, avoiding possible downstream complications of implementing a completely new paradigm of behavior. The reader is referred to Bishop (2006);

Abu-Mostafa *et al.* (2012) for further details about supervised ML techniques.

This experiment aims to efficiently identify en-route TVs likely to be regulated for a given time interval. Concretely, the work focuses on *C-ATC Capacity ATFM regulations* in TVs from both the Maastricht Upper Area Control Centre (MUAC) and REIMS [1] regions. The reason for focusing on this particular type of regulation is because, as seen previously in Table I-2, they are the most frequent type of regulation reason. Regulations tagged as C-ATC Capacity indicate that the network was operating under normal conditions, but the expected demand was above the capacity. On the other hand, the analysis focuses on the MUAC and REIMS regions because they are two of the most regulated areas in the European airspace (PRC, 2019, 2021). Therefore, if the selected case studies reported good performance, it could indicate that the methodology could be extended to less congested regions as they should be less challenging regions. Appendix A provides additional results for a less challenging TVs from Spain to show the scalability ability of the proposed framework.

Figure III-1 is a simple representation of the intention behind the presented experiment, where the colored ATC sector represents a congested region that should be regulated.

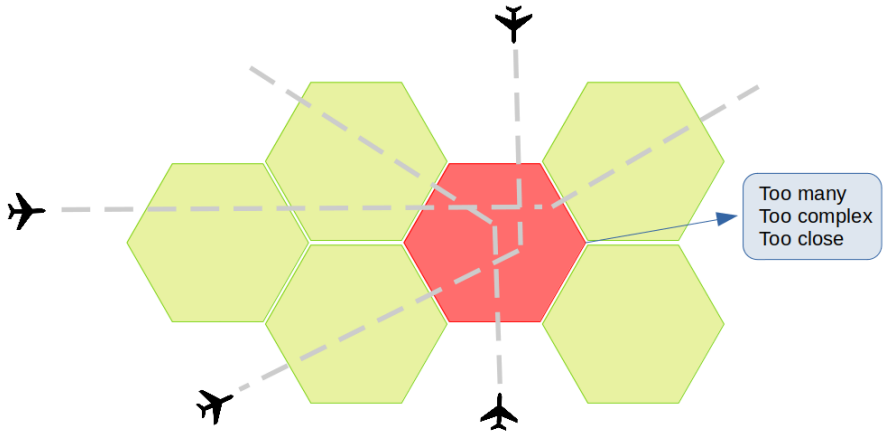



**Figure III-1:** *Visual abstract identification ATFM regulations at TV level.*

### III.2.1 Assumptions

An inherent assumption in supervised learning is that the noise in input features and labels is low. In our case, it is assumed that the data in the datasets (Aeronautical Information Regulation and Controls (AIRACs)) are accurate and that the decision to apply a regulation was correct. However, depending on the exact pre-tactical time horizon with respect to the day of operation (D0), not all pre-tactical routes may be defined for all the flights. In this case, it is assumed that the NM has the tools to estimate the flight plans for these flights, which is the procedure followed where PREDICT estimates the route of the pre-tactical flights (Niarchakou, 2022). Furthermore, according to some preliminary analysis and Martín Martínez *et al.* (2020), between 83% and 90% of the origin-destination pair always present the same pre-tactical route.

Another usual assumption in supervised ML is that non-modeled features have negligible effects. Many features could be used for the work done in this Chapter, but as it was probed in Gianazza (2010), most of the complex features are strongly correlated with the simple ones. Thus, the simple features are the most representative ones. However, not all possible complexity indicators related to ATFM regulations are used (e.g., number of available controllers), but additional information related to the air traffic complexity is also inherent in the images used.

[1] The REIMS region refer to the airspace region around the city of Reims, located at the northeast of France.

## III.3 Data analysis

This section summarizes the data sources used for this case study (see Section III.3.1) and the Exploratory Data Analysis (EDA) of the labeling (see Section III.3.2).

### III.3.1 Data sources

The data sources required for the development of the case study that predicts ATFM regulations at the en-route TV level are summarized in Table II-3. The AIRACs are used to compute the features related to operational and demand information, while ERA5 is the selected source of weather information. Finally, because the models are expected to be used by the NM, the observations are labeled using data from EUROCONTROL.

**Table III-1:** *Data sources used to predict en-route C-ATC Capacity ATFM regulations (TV level)*

| Data sources / Format | Period time | Usage | Comment |
|---|---|---|---|
| AIRAC | June, July, August, September 2018 | Features | M1 traffic |
| ECMWF | June, July, August, September 2018 | Features | ERA5 forecast |
| EUROCONTROL | June, July, August, September 2018 | Labelling | Boolean |

### III.3.2 Exploratory data analysis

The observations have been labeled according to ATFM regulations cataloged as C-ATC Capacity provided by EUROCONTROL. Figure III-2 shows the number of instances per selected region according to the possible regulation reasons. It can be seen that C-ATC Capacity regulations are the most frequent regulation reason.

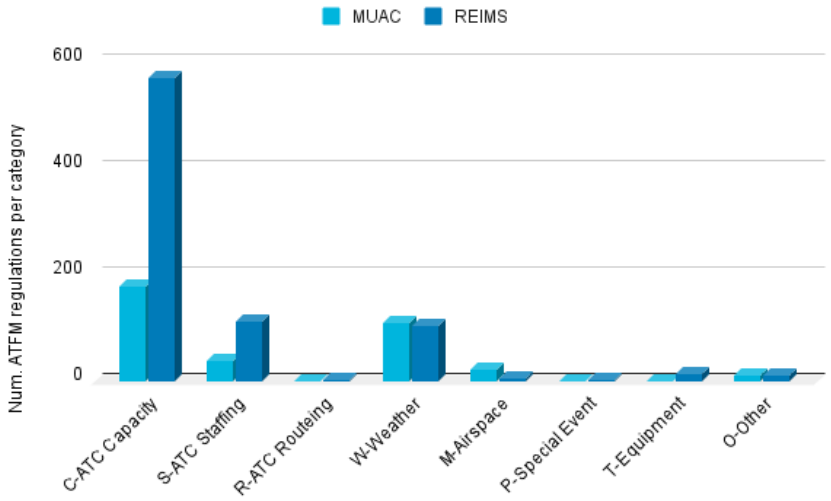



**Figure III-2:** *Number of regulations per category in the available AIRACs*

For the MUAC region, we have 176 C-ATC Capacity ATFM regulations for en-route traffic along 71 different days, a mean number of regulated TVs per day equal to 2.5, and a mean duration per regulation of 122.02 minutes. On the other hand, for the REIMS region, we have 570 regulations for en-route traffic in 96 days, a mean number of regulated TVs per day equal to 5.96 with a mean duration of 101.2 minutes.

As an example, Figure III-3 shows the final regulations for the most regulated TV in the MUAC and REIMS regions during the four months of data available. The regulations from these four months have been stacked and used the color map to show coincidences between days and hours. As can be seen, most of the regulations were implemented between 10 am and 11 am, but they also appear during all the open hours. Notice that similar characteristics are present in other TVs from both regions.

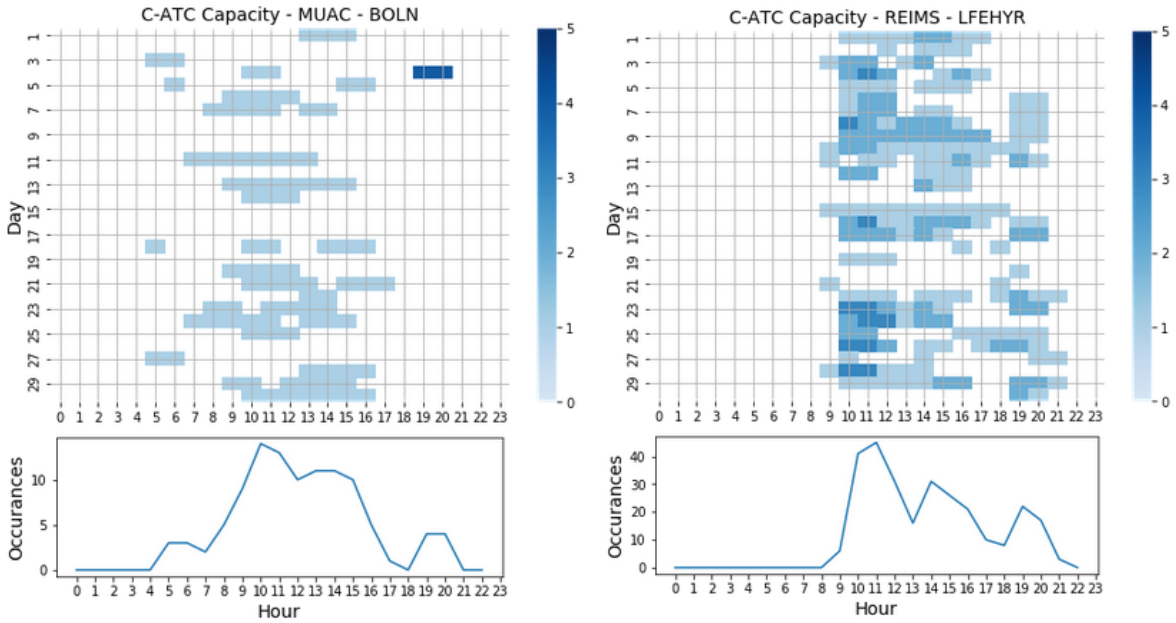



**Figure III-3:** *Heatmap C-ATC Capacity regulations.*
***(Left)** BOLN-MUAC **(Right)** LFEHYR-REIMS*

## III.4 Predictive capabilities

Three different ML models based on a time-distributed approach are studied to predict C-ATC capacity regulations at the TV level to take advantage of different types of information. First, the *RNN-based model* predicts ATFM regulations using scalar variables. Second, the *CNN-based model* identifies regulation using images. Third, *RNN-CNN hybrid models* combining the previous two.

Section III.4.1 particularizes the characteristics of the input observations and input features used for both the RNN-based and CNN-base models; thus, for the RNN-CNN hybrid models. Section III.4.2 presents the outcome of the models and the intention behind them. Section III.4.3 shows the proposed architecture for the RNN-based models. SectionIII.4.4 presents the architecture of the CNN-based model. Section III.4.5 details the different proposed hybrid architectures.

### III.4.1 Inputs of the models

The developed time-distributed frameworks based on artificial NN use two types of inputs. *Scalar variables* for the time-distributed RNN, and *artificial images* for the time-distributed CNN. In both cases, the information is extracted from the AIRACs used in R-NEST to generate samples of 30-minute intervals sliced into one-minute time-steps.

There are four main reasons behind using 30-minute intervals. First, despite being conservative, ATCOs look at a minimum possible interval. Second, the end tool aims to predict possible ATFM regulations for specific intervals of time (e.g., “Is needed a regulation from 8 am to 10:30 am on 28th September?”), and for a given day (e.g., “What are the regulations required for 28th

September?"). Third, the system aims to identify the moment a regulation shall start and ends as precisely as possible. Fourth, the average duration of the regulations in 2018 was 110 minutes. Thus, using 30-minute intervals allows us to show the models a wide variety of input samples: transitions between no-regulated and regulated intervals (and vice versa), purely non-regulated, and completely regulated intervals.

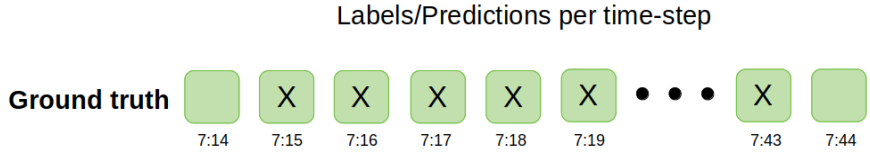



**Figure III-4:** *30-minute interval sliced into one-minute time-steps*

Regarding the input samples, it is used a balanced dataset composed of approximately the same number of positive and negative time-steps. Half of the negative observations were extracted from days without regulations and half from regulated days to help the ML models precisely identify the different ATFM regulations. Furthermore, to ensure that observations in the training set are not used for the testing, from the four available AIRACs, the first three (84 days) are used for training and the fourth (28 days) for testing. This corresponds to the conventional 70–30% split for training and testing. In the end, the dataset used contains approximately 1500 30-minute intervals for the MUAC region and 5000 30-minute intervals from REIMS.

Notice that all the input features are normalized to avoid the vanishing gradient problem, which appears when training artificial neural networks with gradient-based learning methods and backpropagation (Basodi *et al.*, 2020). In such methods, during the training phase, each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight. The problem is that the gradient will be vanishingly small sometimes, preventing the weight from changing its value. In the worst case, this may stop the neural network from further training.

#### III.4.1.1 Scalar variables for the RNN-based model

The input *scalar variables* used for the RNN-based model can be directly exported from R-NEST. The RNN uses a combination of basic features and those presented in Gianazza (2010) as the most representative to exhibit the traffic complexity. The following list shows the scalar variables provided to the model for each of the 30 time-steps that compose an input sample:

- **Interval**: associated 30-minute interval of the studied day (from 0 to 48);
- **Day of the week**: day of the study (from 0 to 6);
- **Capacity of the TV**: sustain capacity of the TV under normal operational conditions;
- **Occupancy Count**: expected number of flights inside the TV for the next 20 and 60 minutes;
- **Entry count**: expected number of flight entering in the TV for the next 20 and 60 minutes;
- **Workload**: expected workload in the TV for the ATCOs;
- **Conflicts**: number of conflicts in the TV;
- **Number of flights at the different phases**: number of flight climbing, cruising, and descending.

Notice that the previous list of scalar variables is provided per time-step in the 30-minute intervals. That is because the model aims to process information that evolves on time to capture the moment the regulation should start and end accurately.

The interval of the day is used because, typically, more regulations are implemented in the morning to avoid the propagation of possible disruptions in the network. Similarly, the day of the week is used to show the model that different traffic levels are expected on different days of the week. Traffic rises from Friday to Sunday. Capacity is provided to the model as an indicator of the amount of considered safe traffic. The occupancy and entry count are metrics directly related to the expected demand. Occupancy count refers to the expected number of flights inside the sector for a specific time interval, while the entry count shows the expected number of flights entering the sector. The expected workload computed by R-NEST is used because it is one of the primary reasons behind the implementation of regulations. Finally, the number of conflicts and flights at different phases are used as indirect indicators of the expected workload. The larger the number of conflicts or flights at different phases, the larger the workload of the ATCOs because more information must be considered.

#### III.4.1.2 Images for the CNN-based model

Each TV has different characteristics, not only in terms of the features used by the RNN-based model but also shape, traffic flows, or entry and exit points. This information is not encoded in the AIRACs; thus, it cannot be directly extracted and used as scalar input features. However, this information can be encoded in images, allowing the ML models to figure it out by themselves. The artificial images are intended to provide additional information related to the complexity of the traffic and scenarios.

Similarly to the RNN-based model, the goal is to develop a CNN-based model able to process images that evolve over time. Therefore, the input samples are sequences of images showing the airspace configuration at consecutive time-steps. Figure III-5 presents the images used by the CNN-based model, where the colors show if the aircraft is climbing, cruising, or descending, the circles express the location, and the lines the heading.

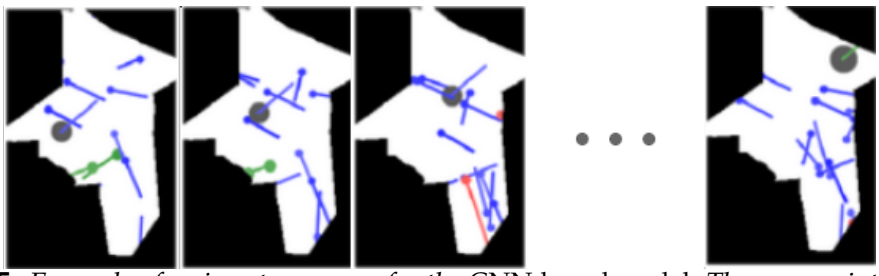


**Figure III-5:** *Example of an input sequence for the* CNN-based model. *The gray points show the path of a unique aircraft. The complete sequence contains 30 images*

The sequences of artificial images are generated using the pre-tactical trajectories available in the AIRACs. The sampling rate of these trajectories varies, often providing more data points (aircraft ID, date/time, latitude, longitude, Flight Level-FL) during the departure and arrival phases than at the cruising phase. By assuming constant speed between data points, we can interpolate the trajectories (see Basora *et al.* (2017) as another example of interpolation), and obtain both the location (latitude, longitude, FL) and heading of each aircraft inside a TV for a particular time-step. Finally, to represent the shape of the TV, from the file *Newmaxo ASCII Region file*, it is extracted the set of pairs (latitude, longitude) that define the perimeter of a TV.

### III.4.2 Outputs of the models

The native output of the time-distributed models is a probabilistic prediction per class and input time-step. For each 30-minute sample sliced into one-minute time-steps, the models produce 30 predictions corresponding to the probability of each time-step being regulated. Figure III-6 evaluates the predictions from the models for an input sample from 7:14 am to 7:44 am. Notice that each square represents a time-step, and the ATFM regulation has been labeled using an *X*.

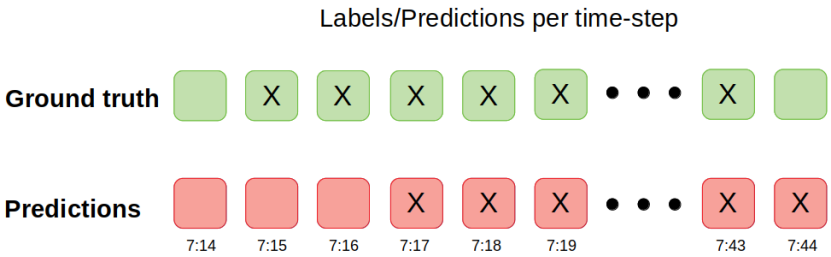



**Figure III-6:** *Example of the outcome from the ML models*

### III.4.3 RNN-based model architecture

The architecture of the RNN-based model is composed of one input layer, two hidden layers, and one output layer. Once the input observations are fitted into the model, they are passed to the hidden layers. Each hidden layer has a time-distributed wrapper which allows the NN to process every temporal slice as input with the same set of weights, each composed of several Long-Short Term Memory (LSTM) cells. The first hidden layer is an LSTM composed of 32 units. The second layer is a Dropout to reduce possible overfitting. Then, the previous two layers are repeated. Finally, the output layer is a time-distributed dense layer that allows the model to make binary predictions. Figure III-7 visually represents the architecture used.

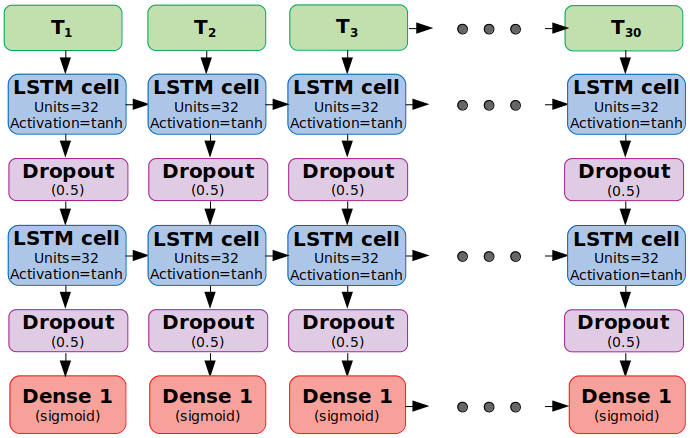



**Figure III-7:** *RNN-based model architecture for en-route C-ATC regulations*

### III.4.4 CNN-based model architecture

The CNN-based model aims to process images that evolve over time. Therefore, the input samples are sequences of images showing the airspace configuration at consecutive time-steps (see Figure III-5). Similar to the previous RNN-based model (see Section III.4.3), the final developed architecture presented in Figure III-8 captures the temporal evolution of the expected airspace configuration since the images per time-step are processed in parallel using the time-distributed wrapper.

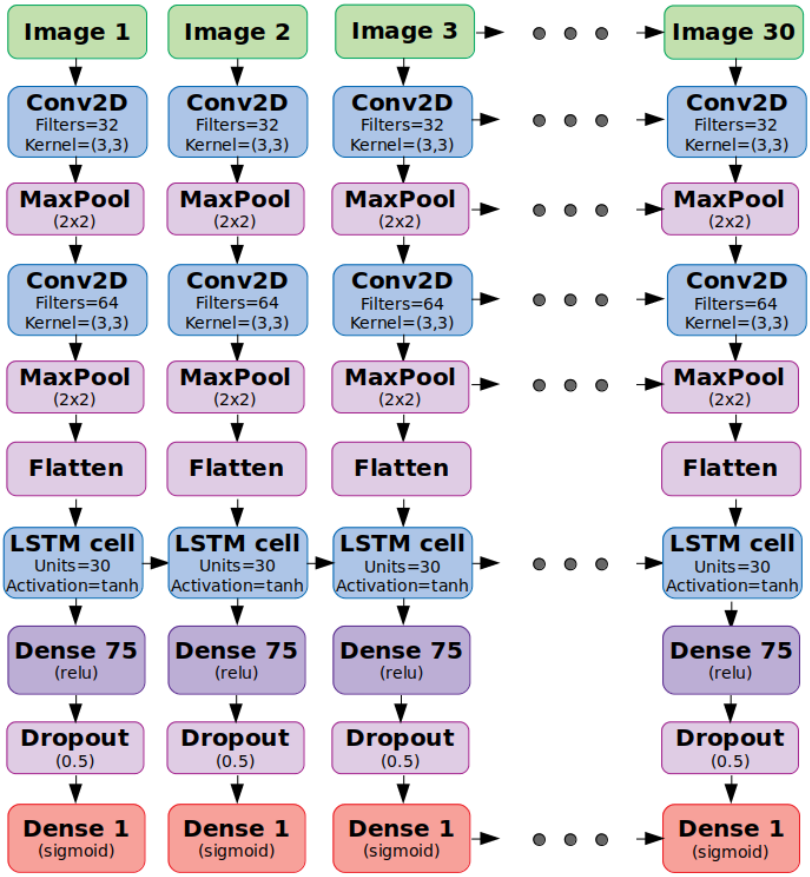



**Figure III-8:** *CNN-based model architecture for en-route C-ATC regulations*

Concretely, the first hidden layer is a time-distributed 2D convolutional layer that uses 32 filters with a kernel size (3, 3) using a Glorat initialization (aka. Xavier initialization). It is followed by a time-distributed MaxPooling layer with a pool size equal to (2, 2). The third layer is a time-distributed 2D convolutional layer with 64 filters and a kernel size equal to (3, 3), followed by another MaxPooling layer. Next, The fifth layer is a Flatten, which reshapes the input tensor to have a shape equal to the number of elements in the tensor. The Sixth layer is an LSTM cell, which captures the temporal information. The seventh is a fully connected layer with 75 neurons. The eighth layer is a Dropout used to reduce overfitting with an activation rate of 0.5. Finally, the output layer is a fully connected layer with one neuron.

### III.4.5 RNN-CNN Hybrid model architecture

The hybrid framework proposed to predict C-ATC Capacity ATFM regulations combines the previous two types of ML models. The RNN-based model processes general metrics based on scalar variables, while the CNN-based model is able to process the specific airspace configuration and the distribution of the airspace traffic. Combining both models could be key to obtaining the best possible performance. Different types of information are processed, as well as the probabilistic outputs, which could reduce false positive and negative predictions.

Three hybrid architectures are investigated in this Chapter. The first approach, depicted in Figure III-9, uses the RNN-based model to extract the relevant information from the scalar variables. Then, the CNN-based model is used to extract the relevant features from the artificial images. Finally, the resulting features are passed through a time-distributed classifier to produce the final prediction.

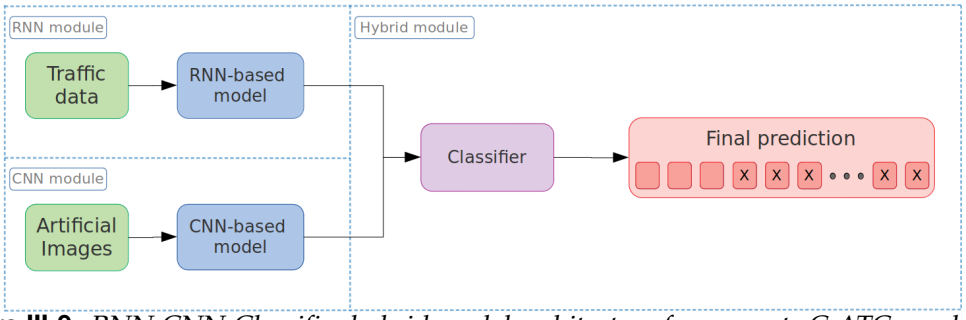



**Figure III-9:** *RNN-CNN-Classifier hybrid model architecture for en-route C-ATC regulations*

The second hybrid model architecture can be seen in Figure III-10. The CNN-based model is used to extract the main features from the images, then they are concatenated with the scalar variables, and the final input sample is processed by the RNN-based model to obtain the final prediction.

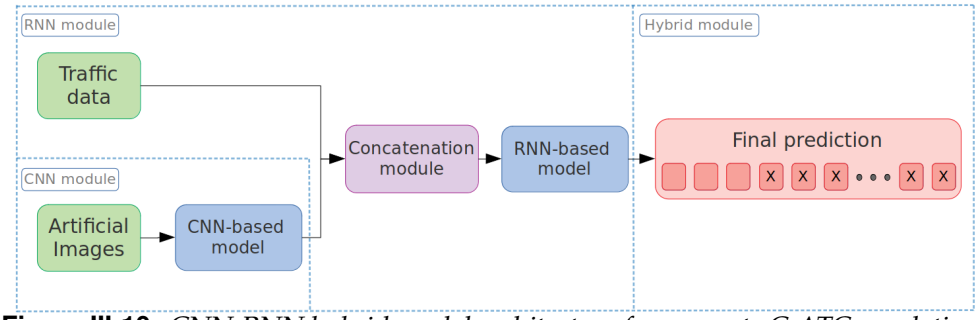



**Figure III-10:** *CNN-RNN hybrid model architecture for en-route C-ATC regulations*

Finally, the third hybrid model architecture is based on a *RNN-CNN cascade architecture*. It starts making predictions on a 30-minute interval using the RNN-based model. If the model has high confidence in the prediction, it will be the final prediction. On the other hand, if the model presents low confidence uses the CNN-based model to refine the initial prediction (see Figure III-11). More precisely, and taking into account the information obtained from the *confidence-level analysis* in Section III.6.1, the CNN-based model is used when the average activation from the RNN-based model is between 0.35 and 0.90. Then, both models' activation values at the output layer are averaged and used to obtain the final prediction. Otherwise, the final prediction only comes from the RNN-based model.

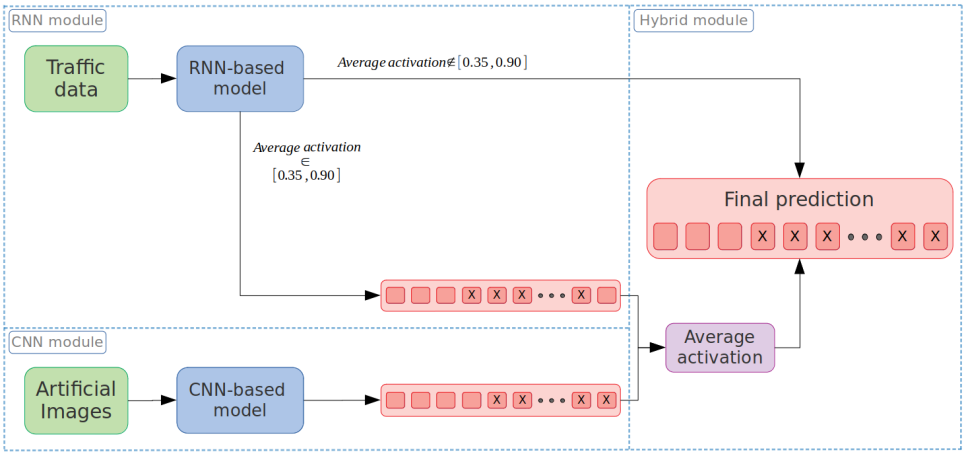



**Figure III-11:** *RNN-CNN cascade hybrid model architecture for en-route C-ATC regulations*

The predictions are refined when they exhibit an average activation between 0.35 and 0.90 because they are considered low-confidence predictions. Correctly predicted no-regulated intervals have an activation close to zero, and correctly predicted regulated intervals have an activation higher than 0.90. Therefore, using the proposed interval, we can re-evaluate intervals where the RNN-based model is not very confident about a possible required regulation, such as intervals containing transitions (from non-regulated to regulated time-steps, and vice versa).

Notice that predicting each input sample with the two models increases the computational time considerably. The predictions from the CNN-based models are more computationally expensive (about ten times slower than the RNN-based model) mainly due to the cost of generating the artificial images.

## III.5 Performance evaluation

This section shows the results of the experiment that predicts C-ATC capacity ATFM regulation at the TV during the pre-tactical phase. Section III.5.1 specifies the evaluation metrics. Section III.5.2 and Section III.5.3 present the results obtained from the individual RNN-based and CNN-based models, respectively. Section III.5 shows the results obtained to select the final hybrid model and the results from the final RNN-CNN cascade model that exhibits the best overall performance.

In all the experiments, results from specialized models for the three most regulated TVs in both the MUAC and REIMS regions are presented. Together with a ML model trained to predict regulations in all TVs composing the previously mentioned regions. Results from a model designed to predict ATFM regulations over the entire region are shown to prove the scalability of the presented architectures. Although the performance of the specialized models is higher, this experiment provides an alternative for those TVs where not enough samples are available for the training. Data availability is key in supervised ML models.

### III.5.1 Evaluation metrics

Because of the nature of the models, the first analysis studies the ability of the models to predict what exact time-steps are going to be regulated. This analysis is called *time-step analysis*. However, a prediction per time-steps could exhibit a too-fine granularity for the current Collaborative Decision Making (CDM) process done in DCB. Therefore, it is proposed a second *interval analysis* where information from the entire input sample is taken into account. Furthermore, as has been mentioned, predicting the exact moment an ATFM regulation shall start and end is challenging.

In both cases, the accuracy, recall, precision, and F1-score are used to validate the performance of the models. At the time-step level, each input time-step is classified as positive (regulation needed) or negative (no regulation implemented). On the other hand, interval classification is based on grouping the models' predictions to determine whether the 30-minutes interval contains a regulation. An interval is considered regulated if the number of positive predictions is above a given threshold. This evaluation sets the threshold to five time-steps for two reasons. First, false-negatives (not detecting a needed regulation) are considered more critical than false-positives (predicting a regulation that is not needed), which can be filtered later by the operator. Second, we want to avoid isolated positive time-steps (misclassifications of the model). Figure III-12 is a visual representation of the grouping process used for the interval analysis.

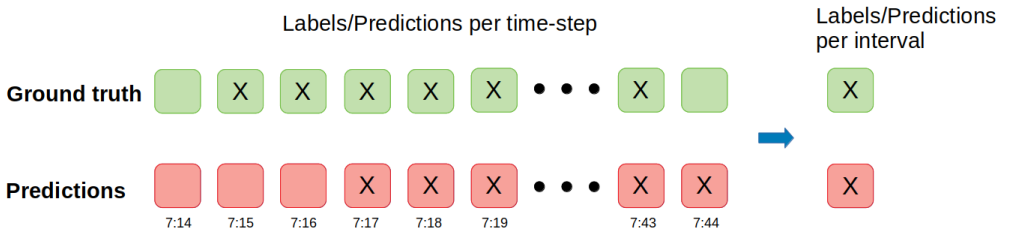



**Figure III-12:** *Time-step VS Interval outputs. Example of the grouping process. (**Left**) Time-steps for a 30-minute interval. (**Right**) Grouped time-steps.*

### III.5.2 RNN-based model

Table III-2 summarizes the results obtained using the *RNN-based model* over the three-most regulated TVs in both the MUAC and the REIMS regions. The results obtained using a single model for the whole region are also included.

From the *time-step* classification results, it can be seen that the specialized models exhibit better overall performance than the models for the entire regions. If we focus on the specialized models, we can see accuracy and recall higher than 80% for all the TVs, and precision between 70% and 90%. In the best case of the MUAC region (BOLN), the model achieves an accuracy of 90.95%, a recall equal to 98.11%, and a precision of 85.51%. The extremely high recall value indicates that nearly all the regulations in this TV are being detected. In the worst scenario (B3EH), an accuracy of 84.14%, recall equal to 92.98%, and precision equal to 70.51% are obtained. The low precision makes this TV worse than D6WH, where an F1-score of 81.75% is obtained versus the 80.82% in B3EH. On the other hand, the best scenario for the REIMS region (LFE5R) exhibits an accuracy equal to 92.46%, a recall equal to 88.82%, and a precision of 91.30%. In the worst scenario (LFEHYR), the accuracy, recall, and precision obtained are 80.06%, 80.31%, and 80.25%, respectively.

When the predictions are made at the *interval* level, all individual TVs in the MUAC region improve all the metrics. However, for the model working over the whole region, despite the improvement in both the accuracy and recall, it presents a 3% drop in the precision (78.57% vs. 75.87%). This is not the case for the REIMS region, where the interval analysis improves overall performance in all the scenarios. Nonetheless, the important aspect of this second analysis is the fact that all the models exhibit a recall equal to 100%. Therefore, they can detect all the 30-minute intervals that contain a regulation.

**Table III-2:** *Performance RNN-based model for en-route C-ATC Capacity regulations at TV level*

| | | | Time-Step Classification | | | | Interval Classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region** | **TV** | **Train/Test** | **Accuracy** | **Recall** | **Precision** | **F1-Score** | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| MUAC | BOLN | 274/119 | 90.95 | 98.11 | 85.51 | 91.38 | 91.51 | 100 | 87.32 | 93.23 |
| | B3EH | 227/100 | 84.14 | 92.98 | 70.51 | 80.82 | 85.54 | 100 | 73.47 | 84.71 |
| | D6WH | 237/107 | 80.04 | 88.82 | 75.73 | 81.75 | 83.22 | 100 | 79.61 | 88.65 |
| | All | 1030/343 | 77.88 | 86.23 | 78.57 | 82.22 | 80.73 | 100 | 75.87 | 86.28 |
| REIMS | LFEHYR | 1061/454 | 80.06 | 80.31 | 80.25 | 80.28 | 88.10 | 100 | 80.29 | 89.07 |
| | LFE4N | 806/348 | 87.21 | 90.97 | 82.69 | 86.63 | 95.36 | 100 | 90.59 | 95.06 |
| | LFE5R | 764/329 | 92.46 | 88.82 | 91.30 | 90.04 | 97.25 | 100 | 93.75 | 96.77 |
| | All | 3670/1573 | 78.29 | 80.52 | 74.82 | 77.57 | 86.97 | 100 | 79.05 | 88.30 |

Last but not least, as an example, to show the proper behavior of the models, Figure III-13 shows the learning curve reported by the RNN-based model used to detect ATFM regulation for the TV MASBOLN in the MUAC region. We decided to present the behavior of the model in this scenario because (a) MUAC is an intermediate region with respect to the number of regulations

available, and (b) MASBOLN is the most regulated TV in this region, and therefore, a challenging TV. Therefore, it is a good representation of the scenarios studied in this work.

As can be seen, the model does not present underfitting. However, there is some overfitting at the end of the training (from epochs 140 to 200). Moreover, some noisy movements can be seen around the validation loss, indicating that the validation dataset is not representative enough of the model's generalization ability. These two drawbacks come from the limited number of samples we have available for the training/testing. Nevertheless, the results obtained indicate that the model is working properly, and these issues can be solved by extending the datasets when more data are available.

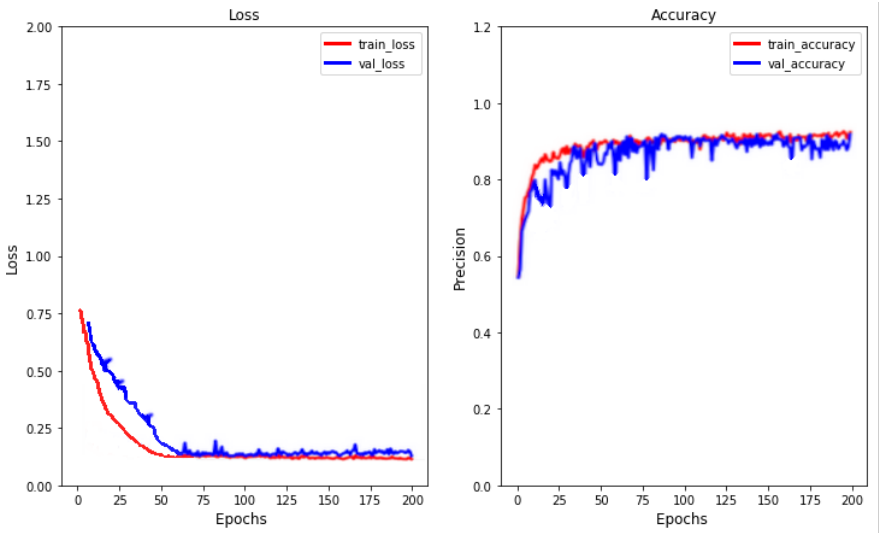



**Figure III-13:** *Training RNN-based model. (**Left**) Loss curve. (**Right**) Accuracy per epoch*

### III.5.3 CNN-based model

Table III-3 shows the results obtained with the CNN-based model. For the *time-step classification*, the specialized models also present better performance than the models for the entire regions. For the MUAC region, the specialized models reported at least an 82% F1-Score, while the model for the entire region exhibits an F1-Score equal to 80.41%. Similar results are obtained for the REIMS region, with a minimum F1-Score equal to 82% in the specialized models and 81.45% for the entire region. If we analyze the accuracy, recall, and precision, it can be seen that for the MUAC region, the best model (BOLN) reported 81.65%, 85.34%, and 82.35%, respectively, while the worse model showed 78.37%, 79.53%, and 82.14%. On the other hand, for the REIMS regions, the best-specialized model reported accuracy, recall, and precision equal to 81.23%, 84.54%, and 85.63%, respectively, while the worse scenario showed 83.57%, 84.23%, and 81.45%.

On the other hand, the *interval classification* presents a higher performance for all the studied TVs across regions. For the MUAC region, a consistent improvement can be seen for the specialized models, with an increase of up to 5% in the F1-Score (BOLN). A similar improvement is obtained in the model for the entire region, where the MUAC region exhibits the biggest improvement, with a 5% increase in the accuracy and up to 14% in the recall. However, it presents a drop of 4% in precision. For the REIMS region, the improvement in the results is also consistent across TVs. Nonetheless, it can be seen that less regulated intervals are detected in REIMS than in MUAC (recall around 85% VS recall around 88%).

**Table III-3:** *Performance CNN-based model for en-route C-ATC Capacity regulations at TV level*

| | | | Time-Step Classification | | | | Interval Classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region** | **TV** | **Train/Test** | **Accuracy** | **Recall** | **Precision** | **F1-Score** | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| MUAC | BOLN | 227/97 | 81.65 | 85.34 | 82.35 | 83.82 | 82.42 | 88.24 | 83.54 | 85.83 |
| | B3EH | 224/96 | 82.55 | 86.67 | 80.02 | 83.21 | 81.78 | 90.03 | 82.49 | 86.10 |
| | D6WH | 226/97 | 78.37 | 79.53 | 82.14 | 81.81 | 80.45 | 85.23 | 83.42 | 84.32 |
| | All | 840/369 | 77.43 | 78.12 | 82.83 | 80.41 | 82.45 | 92.15 | 78.11 | 84.56 |
| REIMS | LFEHYR | 700/300 | 79.54 | 80.56 | 85.63 | 83.02 | 81.54 | 84.13 | 86.13 | 85.12 |
| | LFE4N | 703/301 | 81.23 | 84.54 | 83.63 | 84.08 | 83.56 | 87.73 | 84.79 | 86.23 |
| | LFE5R | 694/227 | 83.57 | 84.23 | 81.45 | 82.82 | 82.82 | 85.32 | 83.23 | 84.26 |
| | All | 2604/1143 | 75.89 | 79.87 | 82.32 | 81.45 | 80.10 | 82.74 | 83.19 | 82.96 |

Finally, Figure III-14 shows the learning curve reported by the CNN-based model used to detect ATFM regulation for the TV MASBOLN in the MUAC region. Similar to the previous RNN-based model (see Figure III-13), the learning curves present some noisy movements indicating that the validation dataset is not ideal due to the limited number of samples available for training/testing. Nonetheless, the results obtained indicate that the model is working correctly.

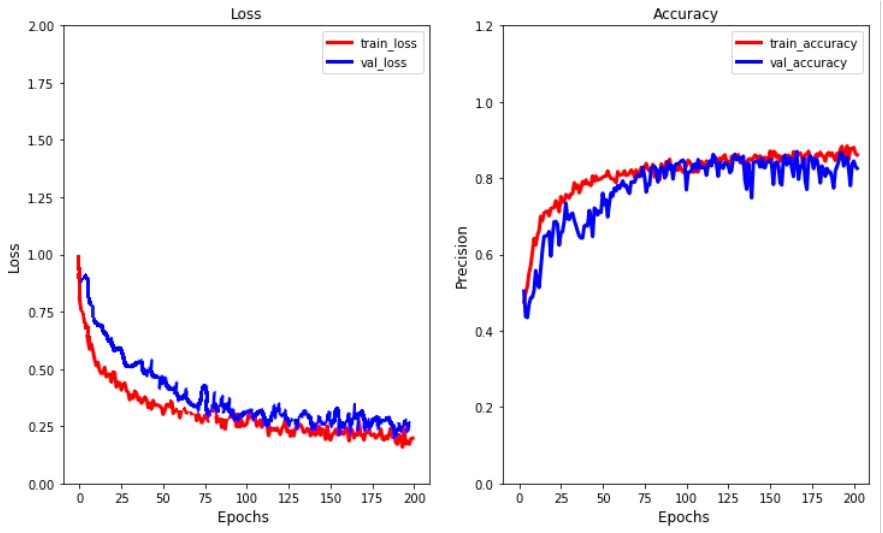



**Figure III-14:** *Training CNN-based model. (**Left**) Loss curve. (**Right**) Accuracy per epoch*

### III.5.4 RNN-CNN hybrid model

Finally, this section presents the results for the three hybrid models that use the previous RNN-based model and CNN-based model. First, a general comparison between the performance of the three models is conducted to select the final approach. Second, an extended analysis of the best hybrid model, the *RNN-CNN cascade* model, is presented.

Figure III-15 shows the average recall, precision, and F1-Score reported by the three studied architectures. The *RNN-CNN-Classifier* shows the results obtained by the hybrid model that uses the RNN-based model to process the scalar variables, the artificial images are processed by CNN-based model, and a third classification model is used to produce the final prediction. *CNN-RNN* corresponds to the hybrid model that uses a CNN-based model to extract the main features from the images, then they are concatenated with the scalar variables, and the final input sample is processed by the RNN-based model to obtain the final prediction. *Cascade* refers to the hybrid model based on a cascade architecture.

The RNN-CNN-Classifier presents a good performance, with a higher recall than precision. Moreover, its computational time is one of the largest because the images for all the observations must be created. CNN-RNN is the hybrid model with the worst performance, probably because the predominant model is the CNN-based model with the worst overall performance. Furthermore, it also requires the creation of all artificial images. Finally, the RNN-CNN cascade model has the best overall performance, probably because the predominant model is the RNN-based model. Therefore, we will focus on the results of the hybrid model based on the RNN-CNN cascade architecture.

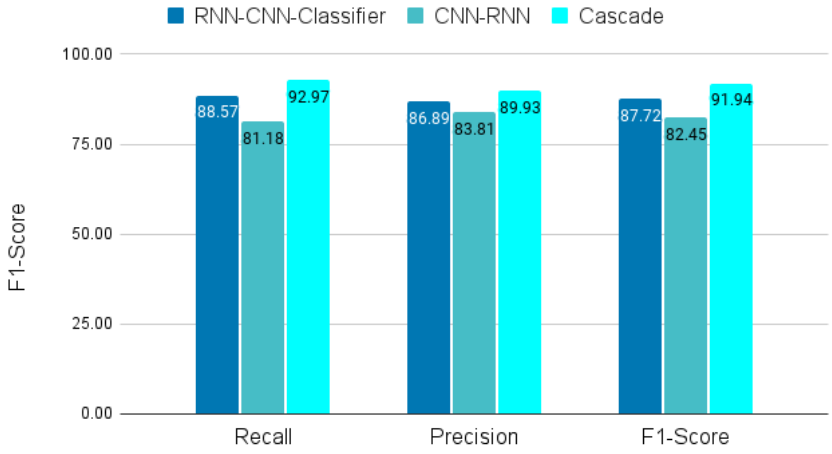



**Figure III-15:** *Average recall, precision, and F1-Score exhibited by the hybrid models. (**Left**) RNN-CNN-Classifier. (**Middle**) CNN-RNN. (**Right**) Cascade.*

The evaluation metrics of the *RNN-CNN cascade model* are shown in Table III-4. The *time-step classification* analysis exhibits a better performance than the previous individual RNN-based model and CNN-based model in all the studied TVs. This architecture is able to improve precision by up to 4% on average, which has the weakest parameter. In the best scenario (REIMS-LFE5R), it exhibits a 10% improvement.

The *interval classification* analysis shows that with less granularity in the predictions, the performance of the models also improves. The accuracy can improve up to 9% (LFE5R from 92.93% to 98.15%), the recall can be increased up to 15% (LFEHYR from 85.28% to 100%), and the precision increments up to 4% (LFE4N from 87.48% to 91.43%). In all scenarios, this analysis shows an improvement in the overall performance of both the specialized and global models across regions, with an average increment of the F1-Score equal to 5%. Moreover, the results show that all the regulations are detected because the recall is equal to 100%.

**Table III-4:** *Performance RNN-CNN cascade model for en-route C-ATC Capacity regulations at TV level*

| | | | Time-Step Classification | | | | Interval Classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region** | **TV** | **Train/Test** | **Accuracy** | **Recall** | **Precision** | **F1-Score** | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| MUAC | BOLN | 376/161 | 91.84 | 98.56 | 86.15 | 91.94 | 92.18 | 100 | 88.78 | 94.06 |
| | B3EH | 260/112 | 87.56 | 93.76 | 81.18 | 87.02 | 88.39 | 100 | 82.74 | 90.55 |
| | D6WH | 289/123 | 85.54 | 90.14 | 85.26 | 87.63 | 85.92 | 100 | 86.43 | 92.72 |
| | All | 1050/450 | 79.94 | 85.89 | 84.76 | 85.32 | 82.56 | 100 | 85.92 | 92.43 |
| REIMS | LFEHYR | 1061/454 | 84.67 | 85.28 | 88.34 | 86.78 | 89.78 | 100 | 88.43 | 93.86 |
| | LFE4N | 806/348 | 88.21 | 91.68 | 87.48 | 89.53 | 97.36 | 100 | 91.43 | 95.52 |
| | LFE5R | 764/329 | 92.93 | 93.54 | 92.78 | 93.16 | 98.15 | 100 | 93.97 | 96.89 |
| | All | 3670/1573 | 80.26 | 83.97 | 81.35 | 82.64 | 87.58 | 100 | 82.49 | 90.40 |

## III.6 Model Explainability

Results from a specific ML model developed to predict regulations over the TV D6WH from the MUAC region are going to be displayed. There are two reasons behind this decision: First, it is one of the models with worse performance; therefore, it is expected to obtain better results for the other TVs. Second, D6WH belongs to the MUAC region, which is the one with fewer training samples. Remember, the success of the DL models mainly depends on the quality and quantity of the input data. These two reasons make D6WH one of the most interesting sectors for the study. Similar results have been obtained for the other TVs, independently of the region.

### III.6.1 Confidence-Level analysis

Figure III-16 shows the confidence-level analysis from the RNN-based model. It shows the number of instances per prediction type as a function of the activation value from the last layer in the NN. As can be seen, the RNN-based model is able to clearly detect the TN time-steps (activation lower than 0.5), showing a small tail between 0 and 0.1. For the TP cases, the behavior is very similar, presenting a small tail between 0.9 and 1 but having the largest grouping close to 1. On the other hand, there is a small accumulation around zero and 0.5 for the FN cases. The accumulation around zero indicates that the model cataloged some time-steps as non-regulated with high confidence, but they should be predicted as regulated. The accumulation around 0.5 shows that for a certain amount of time-steps the model was not sure about being required a regulation or not. For the FP cases, it can be seen a larger accumulation between 0.5 and 0.7 than from 0.7 to 0.9. This indicates that, although the model reports a considerable number of FP, it was not very confident about the prediction in most of the cases. Finally, there is a considerable accumulation between 0.9 and 1, where the model incorrectly predicted a regulation with high confidence. Nonetheless, it is important to notice that the occurrences of the FN prediction are smaller than the ones for the FP cases, indicating that the model is more likely to predict a regulation.

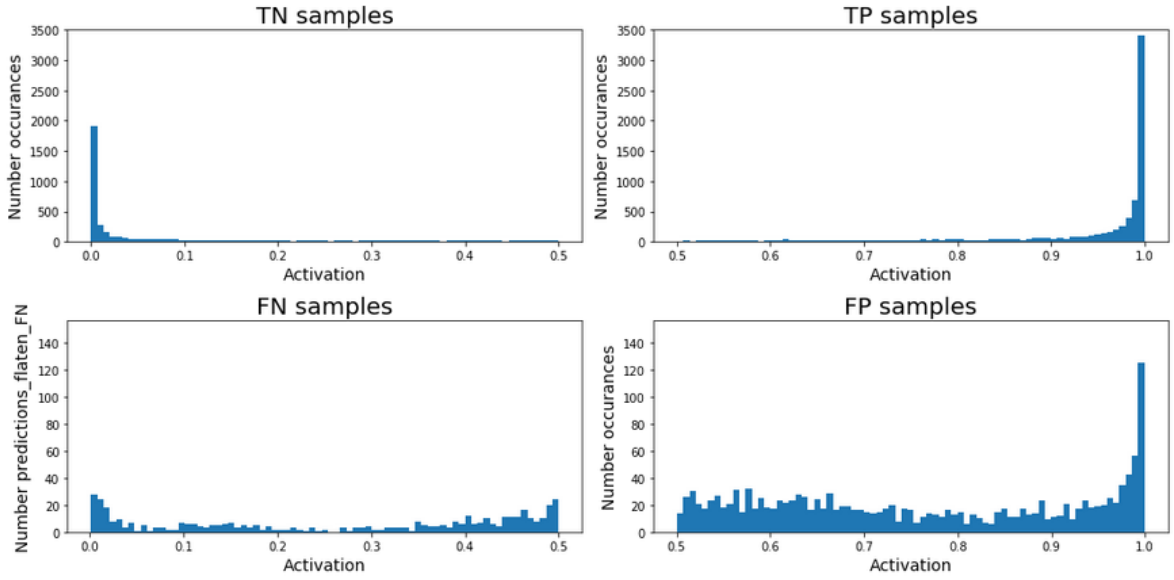



**Figure III-16:** *Confidence-level analysis RNN-based model for TV D6WH predicting en-route C-ATC Capacity ATFM regulations*

The Confidence-level analysis for the CNN-based model can be seen in Figure III-17. For the FN cases, the model presents a tail between 0 and 0.1, with an accumulation of values close to zero, and the TP cases present a tail between 0.85 and 1. On the other hand, a continuous pattern

of behavior can be seen for the FN predictions, with an average value of occurrences under 20. Finally, the FP cases also show a consistent pattern of behavior across activation values between 0.5 and 0.95, presenting a peak between 0.95 and 1. Nevertheless, the number of both FN and FP are smaller compared with the TN and TP cases.

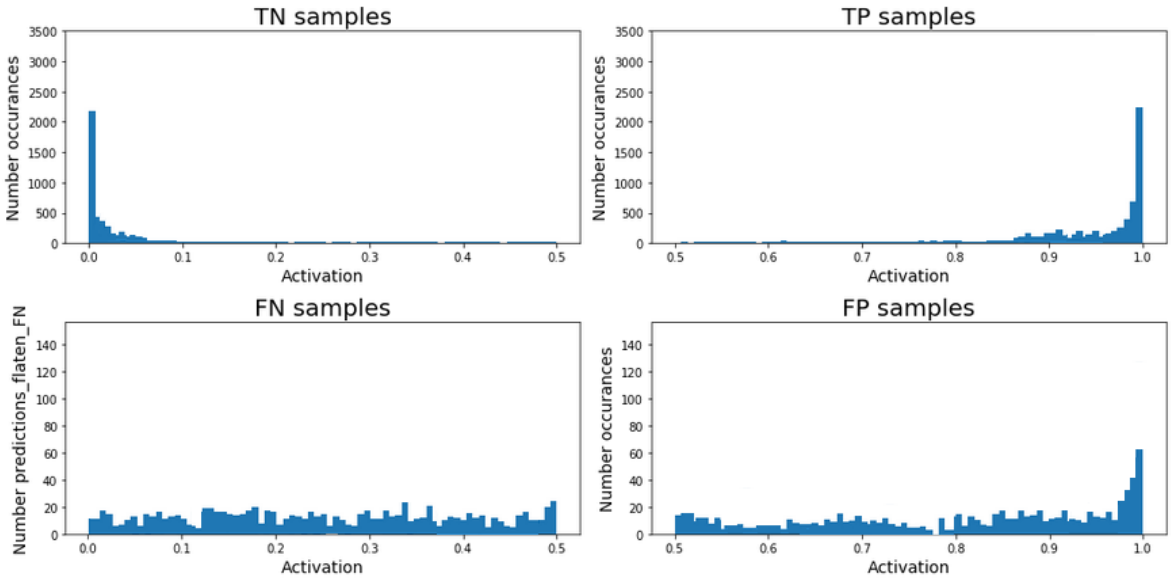



**Figure III-17:** *Confidence-level analysis CNN-based model for TV D6WH predicting en-route C-ATC Capacity ATFM regulations*

Finally, the results for the RNN-CNN cascade model are presented in Figure III-18, exhibiting the most significant accumulation of TN around zero, with a tail between 0 and 0.1. Regarding the TP predictions, the model also presents the largest accumulation between 0.9 and 1. Note that there are almost no occurrences for the rest of the possible values. On the other hand, it can be seen that only a few occurrences are cataloged as FN cases; therefore, the majority of regulated time-steps are identified by the model. If we analyze the FP cases, there are around ten occurrences across all the possible activation values, with a slightly higher peak close to 1.

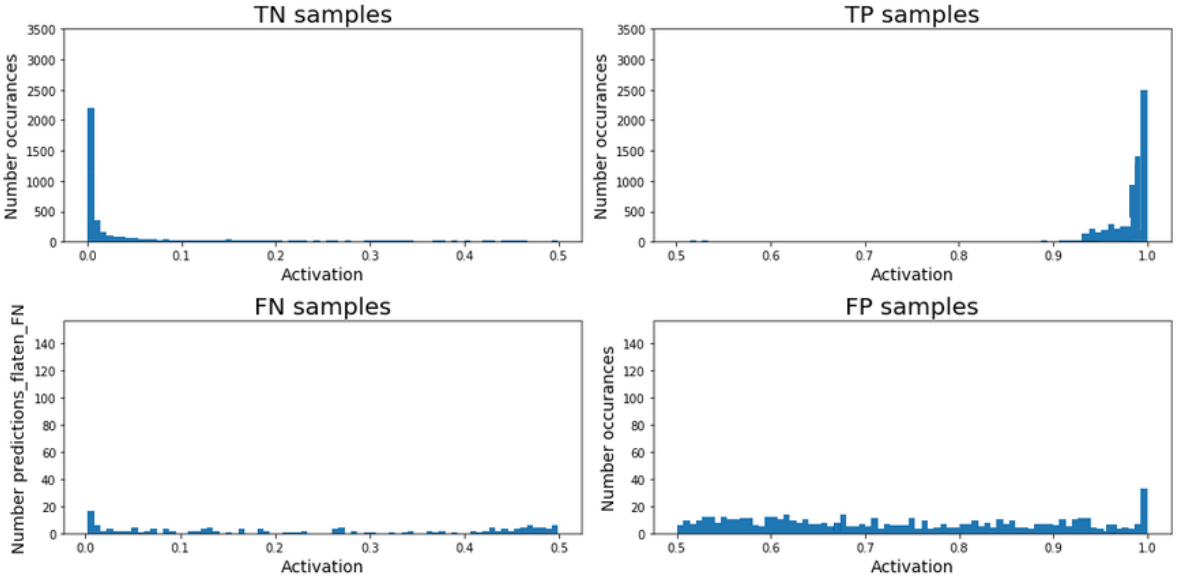



**Figure III-18:** *Confidence-level analysis RNN-CNN cascade model for TV D6WH predicting en-route C-ATC Capacity ATFM regulations*

Let us now numerically compare the results between the RNN-based model and the RNN-CNN cascade model. We can see a 4.3% reduction in the false-negative predictions (from 8.4% to 4.1%) and a 21.2% for the false-positive cases (from 39.2% to 18%). Therefore, the RNN-CNN cascade model presents higher confidence in the predictions and reduces misclassifications.

### III.6.2 SHAP Analysis

The SHapley Additive exPlanations (SHAP) analysis aims to provide information about what input features are more relevant for the models or which ones have a more significant impact. It is important to notice that it is designed to study conventional ML models; thus, it can only be used to study the RNN-based and CNN-based models. As in the previous analysis for model explainability, results from the TV D6WH are presented because it is the individual model with the lowest performance in the MUAC region, being expected better results from the other studied TVs. Notice that the results can be extrapolated to the REIMS region with very similar behavior.

Figure III-19 shows the SHAP values for the RNN-based model. The image shows, from top to bottom, the more relevant input features. The color map indicates how larger or smaller the value of the input feature was, and the location in the corresponding horizontal line represents the activation it generated. The zero in the *X-axis* represents no contribution to the prediction.

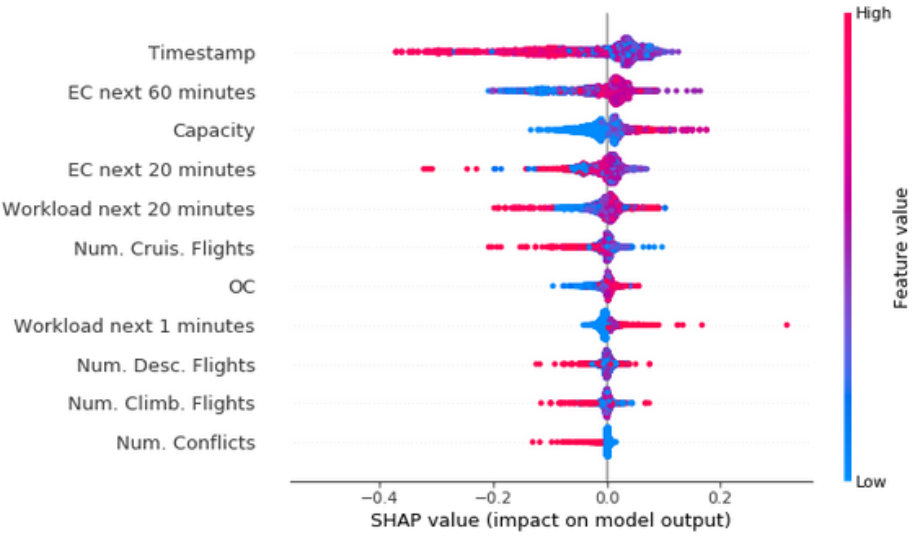


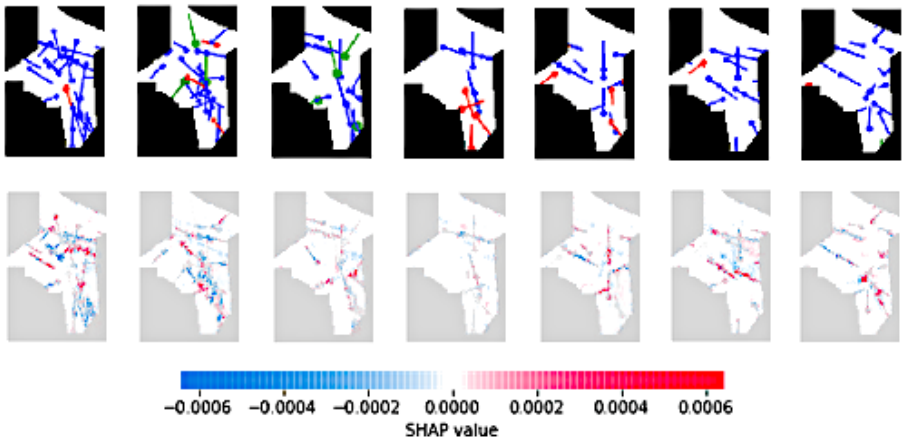

**Figure III-19:** *SHAP values RNN-based model en-route C-ATC Capacity regulations TV D6WH*

The analysis presents the *Timestamp* feature as the most relevant, where samples with a smaller Timestamp (early hours of the day) are more likely to contain a regulation. The second most relevant feature is the *Entry Count for the next 60 minutes*, where larger values produce a higher activation. Therefore, the complexity will increase if more aircraft enter the sector, and regulation is likely required. This is also the case for the *Capacity*, where larger values produce a higher activation. The higher the capacity, the larger the sector; therefore, more aircraft are more likely to generate an overload. The fourth and fifth most relevant features are the *Entry Count for the next 20 minutes* and *Expected workload for the next 20 minutes*, which do not present a clear pattern of behavior. The reason could be that they are relevant features but in combination with another one. From *Number of cruising flights*, it can be seen that small values produce a higher activation. The *Occupancy Count* and the *Workload for the next minute* show the opposite trend. The *Number of descending flights* and the *Number of Climbing flights* do not present an explicit behavior, which is surprising because flights in these two phases should be relevant. The fact that the model cannot properly process these features could be why this TV presents a worse performance. This is also the case for the *Number of conflicts*, where larger values are creating a smaller activation.

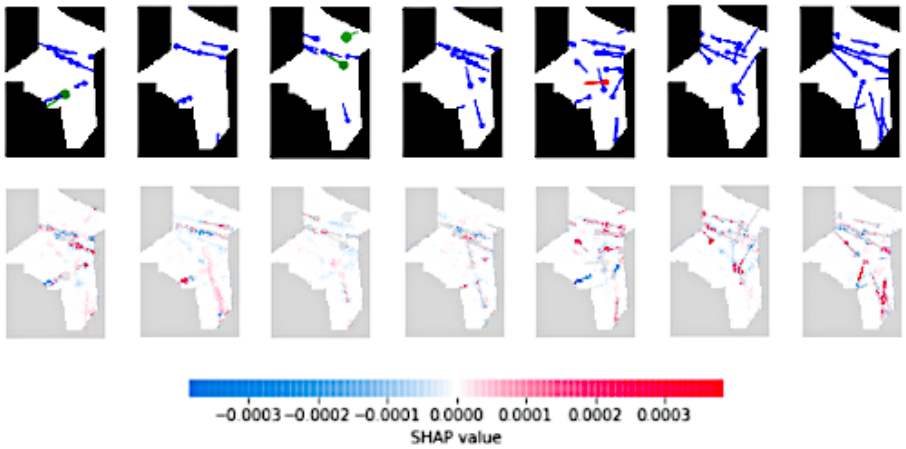
Figures III-20 presents the SHAP values for the CNN-based model. In this case, the analysis highlights the contribution of each pixel to the final prediction of a model. Therefore, it indicates what parts of the image are more (or less) relevant for the models when identifying C-ATC Capacity ATFM regulations. Notice that for clearness, only one picture is shown every five minutes

In the case of the positive prediction (see Figure III-20(a)), it can be observed that a considerable number of aircraft cross the images, creating a larger activation of those flights close in space with the same heading, or flights relatively close in space but with perpendicular headings. It is also interesting to see that the other main source of information comes from flights close to the border of the TV (see the sixth and seventh images), indicating that flights entering or exiting are more relevant to identify possible regulation.

On the other hand, in the case of the negative prediction (see Figure III-20(b)), it is interesting to see a similar pattern of behavior, where aircraft close in space or entering/exiting the TV are more relevant for the model. However, this information is lower, with a Maximum SHAP value of 0.0003. Nevertheless, the model seems to pay attention to the interaction between the two main flows of the TV: one horizontal, in the top part of the TV, and another from the top-left to the bottom-right corners. If we analyze the second image, the top flow indicates that it is not required a regulation. However, the analysis indicated that the model takes more into account the diagonal flow. This could indicate that, even though there is currently no traffic, it is probably expected an increment, which is why it can be seen as a "red line" without any aircraft.




(**a**) Positive prediction




(**b**) Negative prediction

**Figure III-20:** *SHAP values CNN-based model en-route C-ATC Capacity regulations TV D6WH*

## III.7 Advice capabilities

Two advice generators are proposed for predicting C-ATC Capacity regulation at the TV level. First, it is presented a web application that relies on the visualization of the open scheme and the likelihood of regulation along the day of study. The second approach aims to convert the trained models to C++ to be integrated into R-NEST, a model-based simulation tool dedicated to research activities developed by EUROCONTROL.

### III.7.1 Web application for DCB

As a proof of concept, the first advice generator proposed to visualize the predictions of the ML models is based on representation fidelity (Burton-Jones & Grange, 2013), which improves users' understanding of the domain being represented. The idea is to create a web application that emulates the current tools, where the user can easily see when a TV is expected to be operative and the predictions from the models. Furthermore, the tool shall provide information about the uncertainty of the predictions to ensure it provides meaningful advice.

The system is composed of a form (see Figure III-21) and a visualizer (see Figure III-22). The user employs the form to specify the case study, choosing the following parameters:

- **Region**: of interest (MAUC or REIMS)
- **Traffic Volume**: ID of the traffic volume inside the region
- **Day**: of the study
- **Metric**: to analyze in the visualizer
- **Start**: which is the initial desired timestamp
- **End**: timestamp for the analysis
- **Time delta**: which indicates the granularity in the visualizer

The form aims to provide a flexible and user-friendly interface. All the requested information is selected from predefined lists to avoid an incongruent selection. For instance, the region can be MUAC or REIMS (the two regions studied in this Chapter), or some of the metrics available are the Occupancy Count (OC), Entry Count (EC), and complexity. To provide an interactive visualizer, the user can change the zoom (granularity) by selecting a different start, end, or delta.

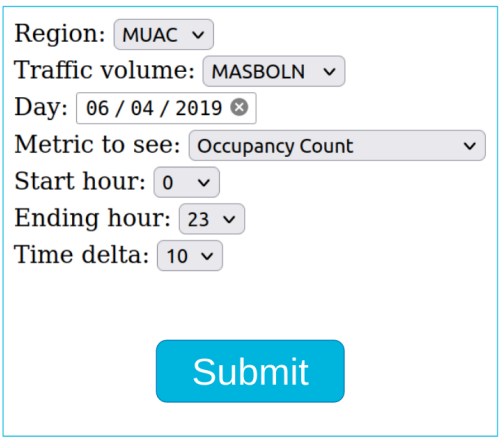



**Figure III-21:** *Advice generator form to select the input parameters*

On the other hand, the visualizer, which can be seen in Figure III-22, shows the outcome of the models. A color scheme is used to distinguish positive predictions with high and low uncertainty and negative predictions:

- **Red**: represents time-steps with demand-capacity imbalance with high probability (probability $> 0.75$);
- **Orange**: indicates time-steps with demand-capacity imbalance with uncertainty ($0.25 \leq$ probability of regulation $\leq 0.75$);
- **Green**: shows the predicted time-steps without demand-capacity (probability $< 0.25$).

The simple proposed color scheme is recognized around the world as a problem (red), warning (Orange), and ok (green), ensuring the web application could be used and integrated around the world.

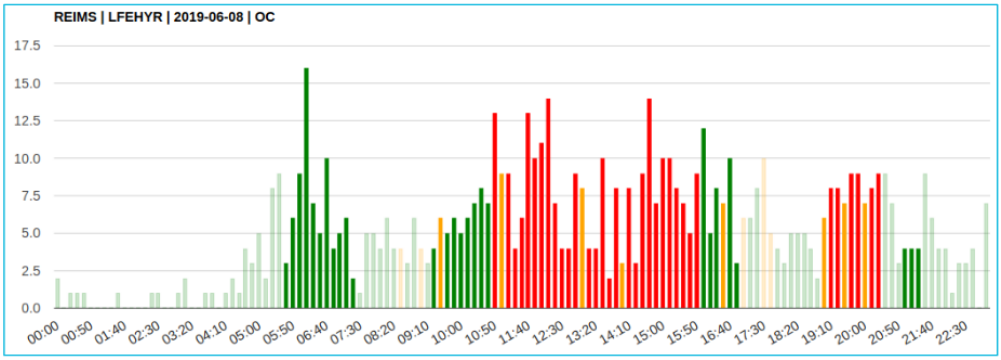



**Figure III-22:** *Advice generator outcome for October 6th 2018, from 0 am to 23 pm*

Notice that a transparency effect has been used to ensure that the advice generator provided insightful information for those intervals of time when the TV is expected to be operative according to the opening scheme.

### III.7.2 Integration into R-NEST

At the moment of writing this thesis, the integration of the ML models into R-NEST is still in progress. However, this section presents the milestones achieved.

First, coordination activities have been done with EUROCONTROL to establish the initial steps of the integration. As a starting point, all partners agree, as a proof of concept, to integrate the *RNN-based model* presented in this Chapter. The models specialized on specific TVs for both the MUAC, REIMS, and Spain (see Appendix A) regions have been converted using the Application Programming Interface (API) *frugally-deep*, as well as the models able to identify regulations over the entire regions.

After converting the models into C++, it is required to start the validation activities. Precompute sets of input features have been adapted to be loaded using C++ to validate the performance of the models, and it has been guaranteed that the outcome of the models is exactly the same. This has been done for approximately 200 different samples. Notice that EUROCONTROL is in charge of developing the required code to compute the input features inside R-NEST as they are the owners; thus, external validation has to be conducted in the future.

A first release ready to be integrated with the performed activities is available, with the converted models and the corresponding documentation.

## III.8 Discussion

The current process to decide whether ATFM regulations are required due to demand-capacity imbalances is a time-consuming task, mainly dependent on previous knowledge and skills of the FMPs and NM operators. Furthermore, despite the variety of tools and metrics available, ATFM regulations are responsible for most of the delays and, consequently, cost. However, future ATM aims to increase airspace capacity between 5% and 10% in the following years, increasing the levels of automatization and information sharing.

Consistent results across different studies and regions have been obtained. The *Time-step classification* analysis reported that the *CNN-based model* performs slightly worse than the *RNN-based model*, with an average F1-Score equal to 83% and 86%, respectively. This drop in performance is especially significant in accuracy and recall. However, the *CNN-based model* exhibits an increment between 3% and 10% in the precision for most of the models (especially in the weakest ones). However, the proposed *RNN-CNN cascade model* benefits from the best of the previous models. It reports accuracy between 85% and 93% for the specialized models, a recall around 90%, and precision between 81% and 94%. In other words, the *RNN-CNN cascade model* exhibits a higher accuracy, recall, and precision than previous models in all the studied TVs.

For the *Interval classification* analysis, a similar behavior pattern is observed. The *CNN-based model* exhibits an overall slightly worse performance than the *RNN-based model*, and the *RNN-CNN cascade model* exhibits the best results. Nonetheless, the most significant result is that the *CNN-based model* cannot identify all the 30-minute regulated intervals. A recall between 80% and 90% is obtained while the *RNN-based model* and the *RNN-CNN cascade model* are able to identify all the regulated intervals. Therefore, the final framework can detect all the intervals that contain an ATFM regulation.

The good results indicate that the proposed approach could improve the current CDM process linked to the detection of demand-capacity imbalances, reducing the overall workload. Furthermore, it has also been found that the *Model explainability* analysis exposes a behavior close to the current CDM procedure. The SHAP analysis has shown that the expected incoming traffic, in combination with the capacity of the TV and the current occupancy, are key aspects to be considered when predicting C-ATC Capacity ATFM regulations. Moreover, aircraft close in space are prioritized, together with aircraft entering or exiting the TV. The transfer of aircraft is one of the main reasons for demand–capacity imbalances due to its complexity.

However, it is worth mentioning that the results obtained maybe are too optimistic. It will be interesting to retrain the models avoiding down-sampling the train/test datasets and comparing the performance between the two approaches. In this scenario, it will be necessary to consider the unbalanced nature of the datasets, *i.e.*, the number of non-regulated and regulated minutes, and implement the required modification. Similarly, utilizing M1 traffic, despite being the closest traffic to the pre-tactical phase, is not optimal because the flight plans are not known. Using tools such as PREDICT[2] would be ideal, but it was impossible due to the limited available data.

Additionally, three major operational constraints are identified in this Chapter. First, despite the excellent performance of the specialized models, the need for models for each TV could produce a scalability issue when deploying a system like this over the entire European network. Perhaps, the specialized models could be used for the most regulated TVs, while the model able to predict regulation for the entire region could be used for the other sectors. Second, a priory the models are designed to predict a specific regulations reason, C-ATC Capacity regulation. Although this is the most frequent type, the other regulation reason should be addressed. Third, the main drawback of this case study is the limited data available to train the models. However, it is worth mentioning that the models have been trained using historical data from the summer,

[2]PREDICT estimates the flight plans when they have not been filled yet, mainly using historical information, comparing the origin and destination, the aircraft type, or the airlines, among others.

which is the season, or one of the seasons, with the largest volume of airspace traffic; thus, the year period with more ATFM regulations. If the models can perform adequately in the most challenging months of the year, they should also show a good performance for the rest of the year.

Also related to the generalization models, one limitation identified in the proposed methodology and approach is that models cannot take into account possible downstream effects of the identified regulations. For instance, the detection and implementation of an ATFM regulation in a particular en-route TV can impact adjacent TVs.

Finally, the need for proper advice capabilities has been shown when developing ML models that aim to be industrialized. Providing meaningful information to the end user is paramount, and more than simply showing the probabilities is required. Two advice generators (a web application and the integration of the ML into R-NEST) are presented based on representation fidelity. Although the tools aim to be used by the NM, the airlines also could take advantage of pre-tactically knowing congested en-route sectors.

# IV

# W-Weather ATFM regulations

In the previous Chapter III, it has been proven that supervised machine learning models can be used to predict C-ATC Capacity Air Traffic Flow Management (ATFM) regulations at the Traffic Volume (TV) level during the pre-tactical phase. Although this was the most frequent ATFM regulation reason in 2018 with a 37.4%, W-Weather ATFM regulations were the second most frequent type with a 25.4% (PRC, 2019). These results are also supported by the Exploratory Data Analysis (EDA) conducted in the previous Chapter (see Section III.3.2).

Convective weather is a well-known aviation hazard; turbulence, wind shear, visibility reduction, lighting, and hail can heavily impact aircraft and Air Traffic Management (ATM). Furthermore, according to climate experts, the frequency and intensity of convective weather will increase in the future (Parodi *et al.*, 2021).

On days with intense convective weather, the airspace conditions are very volatile, directly translating into an increment of the workload of the Air Traffic Controllers (ATCOs). Thus, it implies reducing the airspace capacity to maintain the required safety levels. The reduction in the capacity due to convective weather in conjunction with traffic scheduled to operate in normal conditions typically triggers W-Weather ATFM regulations.

Similar to the previous case study, the introduction of a new support tool for the **detection of W-Weather en-route ATFM regulations** also could reduce the workload, or at least the difficulty, of the Flow Manager Positions (FMPs) and Network Manager (NM) operators during the pre-tactical phase of the Demand-Capacity Balancing (DCB) process. Concretely, this Chapter proposes to adapt the architecture of the previously presented *RNN-based model*, enriching the input features for this new scenario and presenting a model explainability analysis to gain trust in the behavior of the Machine Learning (ML) models.

## IV.1 State of the Art

A significant number of research activities have studied how to translate weather information into appropriate ATFM constraints. Although some works specifically focus on convective weather, this information is considered part of a more complex conjunct in most scenarios.

For instance, ISOBAR project ISOBAR (2020) is a SESAR Joint Undertaking action that aims to use Artificial Intelligence (AI) solutions to Meteo-Based DCB imbalances for network operations planning, identifying demand-capacity imbalances and selecting mitigation measures at local and network level. Similarly, the INTUIT project (Garrigó *et al.*, 2016) used machine learning and visual analytics techniques to identify cause-effect relationships between ATFM regulations and different indicators.

A more narrow approach was presented in Jardines *et al.* (2021), where the authors used regression and classification supervised learning algorithms to predict airspace performance characteristics such as entry count or the number of flights impacted by the regulations for active weather regulation. Kamangir *et al.* (2020) used deep learning Neural Network (NN) and features from numerical weather prediction models to predict thunderstorm occurrence for up to 15 hr(±2 hr accuracy) in advance. Other researchers focus on the consequences and possible preventive actions implemented when facing ATFM regulations due to convective weather. For instance, Marcos *et al.* (2017); Martín Martínez *et al.* (2020) used machine learning and visual analytics to study the airline route choice in the pre-tactical planning phase. Dalmau Codina *et al.* (2019) is a case study for the Maastricht Upper Area Control Centre (MUAC) region, which investigates the predictability of take-off times under adverse weather conditions. Schultz *et al.* (2021); Lattrez *et al.* (2022) studied the weather impact on airport performance through machine learning.

Despite some works in the literature indirectly facing the identification of W-Weather ATFM regulations using ML techniques, to the author's best knowledge, no specific results about the detection of such regulations are available in the literature.

## IV.2 Problem formulation

As shown previously, supervised machine learning models are used to learn patterns from historical data. Therefore, the system aims to learn and replicate past actions in future scenarios. One of the main benefits of using ML techniques is the extremely fast ability of the algorithms to provide advice to the user, making it very suitable for detecting W-Weather ATFM regulations due to their volatility.

Following the good results obtained in the previous case study, this new set of experiments aims to investigate whether the presented **RNN-based model** architecture could be extended to predict a different regulations reason. The reason for focusing on the RNN-based model is that most Numerical Weather Prediction (NWP), as the name indicates, provide weather predictions based on specific Key Performance Indicatorss (KPIs). Combining visual traffic information and radar predictions is required to generate the necessary images for the CNN-based models, considerably increasing the required computation time. Some proofs of concept have been done, but the CNN-based models reported very poor performance in this case study. Therefore, this case study aims to verify whether simple numerical weather features are enough to predict W-Weather ATFM regulations at the TV level.

With this objective in mind, the same input features related to airspace traffic are going to be used because they have proved to be very correlated to ATFM regulations. However, due to the different nature of the problem, it is paramount to enrich them with a set of features related to convective weather. Although the models are trained for the same two regions (MUAC and

REIMS) to ensure a fair comparison of results between the regulations reasons, different TVs have to be selected. For supervised machine learning models, the quality and quantity of data are key aspects directly linked to the performance of the models; thus, the TVs with a larger number of recorded W-Weather ATFM regulations have been selected for each of the regions.

### IV.2.1 Assumptions

The assumptions present in this Chapter are shared with the work done in Chapter III, and they can be summarized as follows:

1. Information in the datasets (Aeronautical Information Regulation and Controls (AIRACs)) is accurate, and the decision to apply or not apply a regulation was correct;
2. It is assumed to have access to PREDICT or a similar tool to estimate any pre-tactical flight plan which might not be defined in the prediction horizon D-1;
3. The traffic characteristics correlated with C-ATC Capacity ATFM regulations are also relevant for W-Weather regulations. Traffic characteristics remain a key component.

## IV.3 Data analysis

This Section summarizes the data sources used for this case study. Section IV.3.1 summarizes the data sources required, and Section IV.3.2 presents the EDA analysis.

### IV.3.1 Data sources

The data sources used for developing this case study are presented in Table IV-1. First, the AIRACs are used to compute airspace traffic features. Second, the numerical weather features are obtained from ECMWF. Finally, labeling the 30-minute samples have been done using a dataset provided by EUROCONTROL; however, it is worth mentioning that equivalent labeling could be obtained from the AIRACs with some data engineering.

**Table IV-1:** *Data sources used to predict en-route W-Weather ATFM regulations (TV level)*

| Data source / Format | Period time | Usage | Comment |
|---|---|---|---|
| AIRAC | June, July, August, September 2018 | Features | M1 traffic |
| ECMWF | June, July, August, September 2018 | Features | ERA5 forecast |
| EUROCONTROL | June, July, August, September 2018 | Labelling | Boolean |

### IV.3.2 Exploratory Data Analysis

For the entire MUAC region, we have 151 *W-Weather* regulations for en-route traffic along 34 different days, a mean number of regulated TV per day equal to 4.42, and a mean duration per regulation of 196.19 minutes. On the other hand, for the REIMS region, there are 582 regulations for en-route traffic in 100 days, a mean number of regulated TV per day equal to 11.7, and a mean duration of 112.5 minutes.

Continuing with the analysis of the available data, it is a combination of weather conditions and the traffic characteristics used to identify weather regulations (individual weather or traffic metrics are not useful). Moreover, traffic characteristics are insufficient to detect possible overload in normal operational conditions. In the MUAC region, if we compare the Occupancy Count (OC) and the Entry Count (EC) metrics with the declared threshold:

- 2.68% of the minutes from regulated periods had an OC higher than the peak threshold, and 2.38% of the minutes had an OC between the sustained and the peak thresholds. On the other hand, for non-regulated periods, 2.53% of the minutes had an OC higher than the peak threshold, and 2.61% of the minutes had an OC between the sustained and the peak thresholds,
- If we analyze the EC for the next 20 minutes for the regulated periods, 14.23% of them were above the peak threshold, and 10.89% of the minutes were between the sustained and the peak thresholds. For the no regulated periods, the analysis showed that 15.1% of the minutes were over the peak threshold, and 12.4% of them were between the sustained and the peak thresholds,
- Finally, analyzing the EC for the next 60 minutes for the regulated periods, 15.35% of the minutes had an EC higher than the peak threshold, and 5.36% of them had an EC between the sustained and the peak thresholds. For no regulated periods, 18.8% of the cases had an EC above the peak threshold, and 10.1% of them were between the sustained and the peak thresholds.

Notice that the results for the REIMS regions are very similar, and only the OC and the EC are analyzed because they are the only ones with predefined thresholds (not all the metrics have an associated threshold or the access to them has not been provided).

As an example, Figure IV-1 shows the regulations for the most regulated TV in the MUAC and REIMS regions along the four months of data available. The regulations from these four months have been stacked and used the color map to show coincidences between days and hours. As can be seen, most of the regulations were implemented between 4 PM and 8 PM. It is worth mentioning that very similar results and characteristics are present in other TVs from both regions.

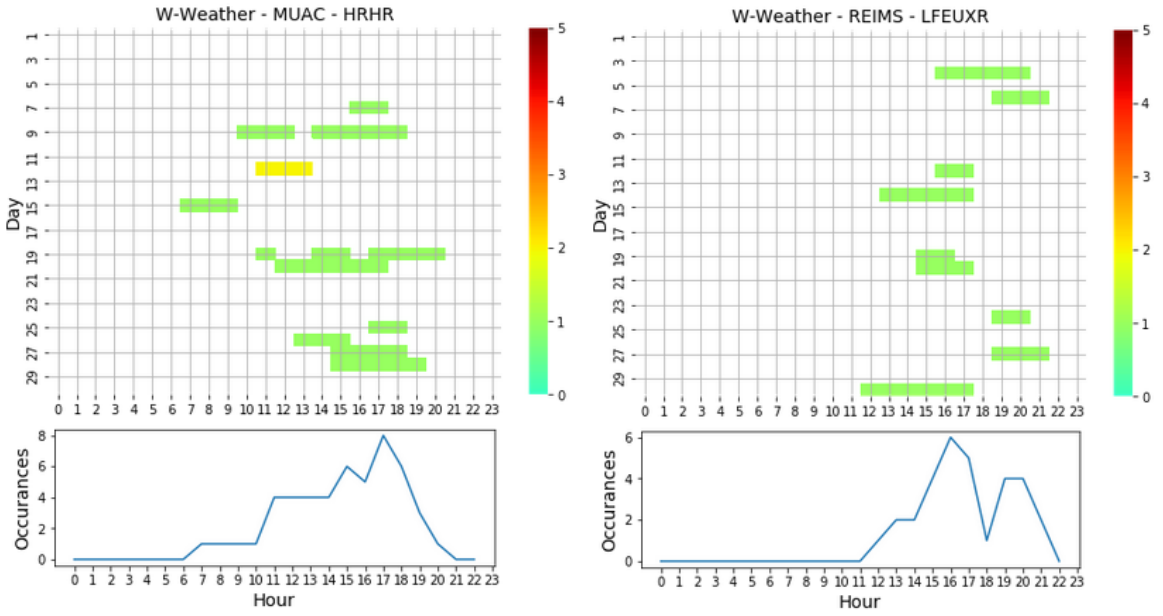



**Figure IV-1:** *Heatmap W-Weather regulations.* ***(Left)*** *HRHR-MUAC* ***(Right)*** *LFEUXR-REIMS*

## IV.4 Predictive capabilities

As mentioned previously, this case study investigates whether the *RNN-based model* architecture presented in the previous Chapter (see Section III.4.3) can be used to predict W-Weather ATFM regulations. Section IV.4.1 presents the new set of input features to predict this regulation reason. Section IV.4.2 presents the adapted architecture.

### IV.4.1 Inputs and Outputs of the model

The architecture of the RNN-based model is based on a time-distributed framework that uses scalar variables as input. The main characteristic of this architecture is the fact that the shape of the input and the output layer of the NN is the same (see previous Figure III-6). 30-minute intervals sliced into one-minute time-steps are used as input samples to obtain 30 different predictions (one per input time-step). As a reminder, 30-minute intervals are used to identify the minimum required interval that has to be regulated, paying special attention to the moment the regulations shall start and end.

To reflect the characteristics of the weather, and having in mind that a key concept is the presence of cumulonimbus (SKYbraby, 2022), the following average values inside the TVs of the following input features per time-step are used:

- **Fraction of cloud cover**: grid boxes covered by cloud (liquid or ice);
- **Potential vorticity**: potential capacity for air to rotate in the atmosphere;
- **Vorticity**: estimated capacity for air to rotate in the atmosphere;
- **Relative humidity**: relative water vapor per kilogram of moist air;
- **Specific humidity**: water vapor per kilogram of moist air,
- **Specific cloud ice water content**: mass of cloud ice particles;
- **Specific cloud liquid water content**: mass of cloud liquid water droplets;
- **Specific rainwater content**: mass of water (aggregated water droplets);
- **Specific snow water content**: mass of snow (aggregated ice crystals);
- **Temperature**: in the atmosphere;
- **u_component of wind**: eastward component of the wind;
- **v_component of wind**: northward component of the wind;
- **Divergence**: rate air spreading out horizontally from a point.

The same combination of input features as in the previous Chapter III is used to express the air traffic conditions. The goal is to validate that the already presented methodology can be extended to predict other ATFM regulation reasons. For completeness, below are the input features used in the previous case study:

- **Interval**: associated 30-min interval of the studied day (from 0 to 48);
- **Day of the week**: of the study (from 0 to 6);
- **Capacity of the TV**: sustain capacity of the TV under normal operational conditions;

- **Occupancy Count**: expected number of flights inside the TV for the next 20 and 60 minutes;
- **Entry count**: expected number of flight entering in the TV for the next 20 and 60 min;
- **Workload**: expected workload in the TV for the ATCOs;
- **Conflicts**: number of conflicts in the TV;
- **Number of flights at the different phases**: number of flight climbing, cruising, and descending.

### IV.4.2 RNN-based model architecture

The proposed architecture to predict W-Weather ATFM regulations is composed of one input layer which receives the input features of the 30-minute samples sliced into one-minutes time-steps. Then, the input samples are passed through three sets of hidden layers, each composed of a Long-Short Term Memory (LSTM) cell and a Dropout layer. All the Dropout layers have an activate rate equal to 0.5, and they aim to reduce any possible overfitting. The LSTM cells are composed of 25, 50, and 25 units, respectively, using a tangent hyperbolic activation function (aka. tanh). Finally, the output layer is a time-distributed fully-connected dense layer that uses a sigmoid activation function to make a binary prediction for each input time-steps. Figure IV-2 is the equivalent visual representation of the architecture.

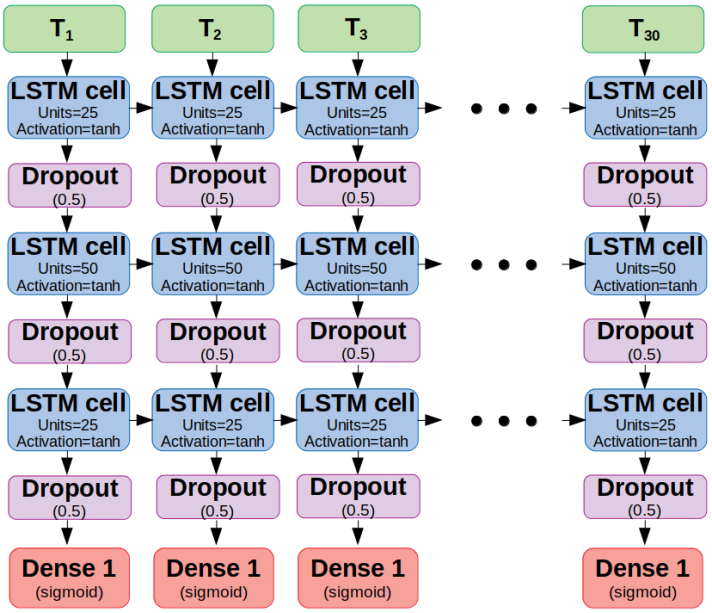



**Figure IV-2:** *RNN-based model architecture for W-Weather regulations*

Similar to the previous Chapter III, the proposed architecture has been obtained after a custom *GridSearch* (Scikit-learn, 2022b) analysis to discover the best configuration. Customizing the analysis is necessary due to the time-distributed wrapper used to take into account information that evolves on time.

## IV.5 Performance evaluation

This Section presents the results predicting W-Weather ATFM regulations using the presented RNN-based model. Section IV.5.1 presents the evaluation metrics and Section IV.5.2 the results.

### IV.5.1 Evaluation metrics

The previously introduced *time-step* and *interval* analyses are performed to evaluate the performance of these models and for consistency across case studies. The *time-step* analysis quantifies the ability of the models to predict precisely the time-steps the TV should be regulated. This analysis is performed by computing the accuracy, recall, precision, and F1-Score. On the other hand, interval analysis is based on grouping the models' predictions to determine whether the 30-minute interval contains a regulation. An interval is considered to have a regulation if the number of positive predictions exceeds five time-steps, reducing possible false-positive and false-negative identification.

### IV.5.2 RNN-based model

Table IV-2 presents the results for the *time-step* and *interval* analyses of the three most regulated TVs in the MUAC and REIMS regions, and the results from models able to predict W-Weather regulations per entire region.

From the *time-step analysis*, it can be seen that for the selected TVs, the models have an accuracy higher than 82%, a recall around 86%, and a precision between 80% and 83%. On the other hand, the *interval analysis* increases accuracy up to 3%, recall 10%, and precision up to 6%. In the best case, recall equal to 97.44 % is reached, showing that the model almost identified all the intervals that contain a regulation. Similar results are obtained in the *matching analysis* where more than 87% of the regulated intervals are precisely detected (perfect and strong matching), allowing the model to have some mismatches.

Results from the REIMS region show a drop around 2%-4% in the precision for the time-step analysis. However, accuracy higher than 80% is obtained in all individual TVs, with a recall between 85% and 88%. The *interval* analysis shows a 4%-8% increment in the precision with a similar recall, indicating higher confidence in the detection of regulated intervals. However, there is a 4% drop in the performance in the *matching analysis* due to being a much more complicated region with a much larger volume of traffic.

**Table IV-2:** *Performance RNN-based model for en-route W-Weather regulations at TV*

| | | | Time-Step Classification | | | | Interval Classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region** | **TV** | **Train/Test** | **Accuracy** | **Recall** | **Precision** | **F1-Score** | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| MUAC | HRHR | 331/142 | 82.43 | 86.52 | 83.67 | 83.13 | 85.42 | 97.44 | 80.14 | 89.32 |
| | HSOL | 234/101 | 84.36 | 86.83 | 82.35 | 84.38 | 87.13 | 89.36 | 84.21 | 87.47 |
| | B3LL | 296/127 | 82.94 | 85.75 | 80.93 | 82.46 | 84.72 | 84.62 | 86.84 | 86.31 |
| | All | 767/328 | 77.84 | 76.37 | 79.88 | 79.44 | 77.3 | 79.67 | 81.43 | 81.28 |
| REIMS | LFE4E | 208/91 | 80.63 | 85.47 | 79.49 | 82.48 | 84.49 | 85.32 | 89.27 | 87.03 |
| | LFEUXR | 208/91 | 87.21 | 90.97 | 82.69 | 86.63 | 89.78 | 82.96 | 88.19 | 85.18 |
| | LFE4N | 206/90 | 83.79 | 86.48 | 78.04 | 80.40 | 80.56 | 88.24 | 94.12 | 91.44 |
| | All | 765/328 | 79.01 | 78.57 | 78.73 | 79.42 | 79.43 | 78.79 | 83.11 | 81.37 |

## IV.6 Model Explainability

To better understand the behavior of the model and gain trust in the predictions, Section IV.6.1 presents the confidence-level analysis, and Section IV.6.2 the results from the SHapley Additive exPlanations (SHAP) analysis.

Results from specific ML models developed to predict regulations over the TV MASB3LL from the MUAC region are going to be displayed. This TV has been selected because the corresponding model shows the overall worse performance; therefore, it is reasonable to expect better results for the other TVs. Similar results have been obtained for the other TVs, independently of the region, which are not shown to avoid redundant information.

### IV.6.1 Confidence-level analysis

The confidence-level analysis presented in Figure IV-3 shows that the model is able to clearly detect the TN time-steps (activation lower than 0.5). There is almost no activation for the majority of these predictions. Moreover, for the TP time-steps, the model shows a high confidence level with an activation higher than 0.9 in most of them (the higher the activation, the more sure the model about regulation is needed). On the other hand, there is a peak around zero for the FN predictions, and from the FP, we can see a small accumulation over 0.9. These mistakes mostly come from the transition between regulated and no-regulated periods. The number of time-steps in these two categories is minimal compared with the TN and the TP, corresponding with the good performance seen in the matching analysis.

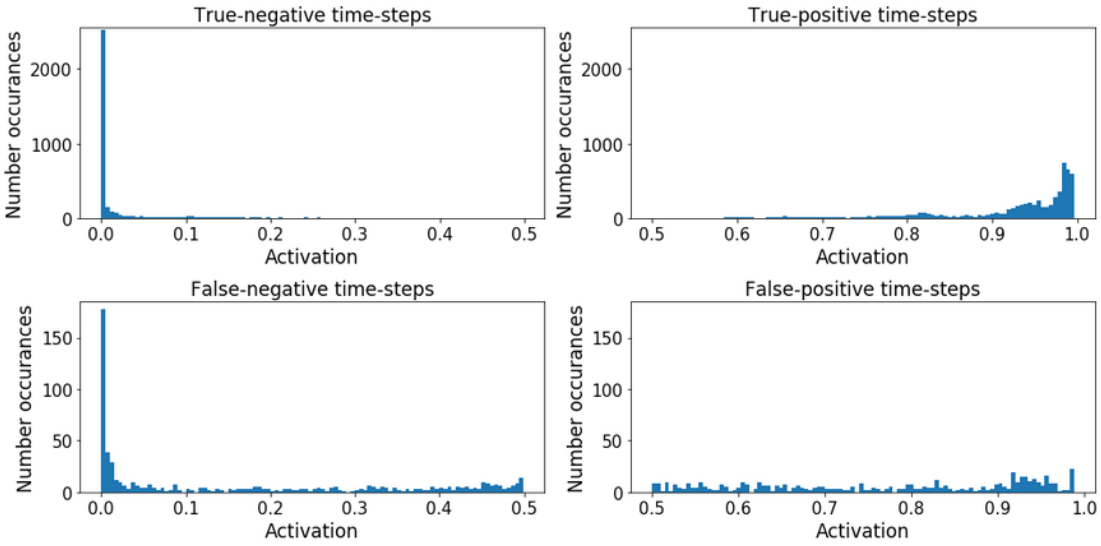



**Figure IV-3:** *Confidence-level analysis RNN-based model for TV B3LL predicting ATFM W-Weather regulations.*

### IV.6.2 SHAP values

Figure IV-4 shows the SHAP analysis for the TV B3ELL in the MUAC regions. The *timestamp* feature shows that more weather regulations are declared at the last hours of the day. Then, the *u-component* and the *v-component* of the wind are the second and fifth most relevant features. However, notice that they show opposite trends. Larger values of the *v-component* produce a higher activation, while small values of the *u-component* create a higher activation. The third most important feature is the *relative humidity*, probably because it is directly related to the amount of cloud and their characteristics. From the *EC for the next 60 minutes*, it can be seen that larger values have a major activation. Nonetheless, the *EC for the next 20 minutes* has an opposite trend, meaning that the expected traffic in a short period is less crucial. This is also the case for the *OC*. In summary, the results indicate that the biggest challenge for weather regulations is a considerable increase in the wind, together with a large expected number of incoming flights. Finally, although the other input features have some impact on the predictions, they are not as decisive.

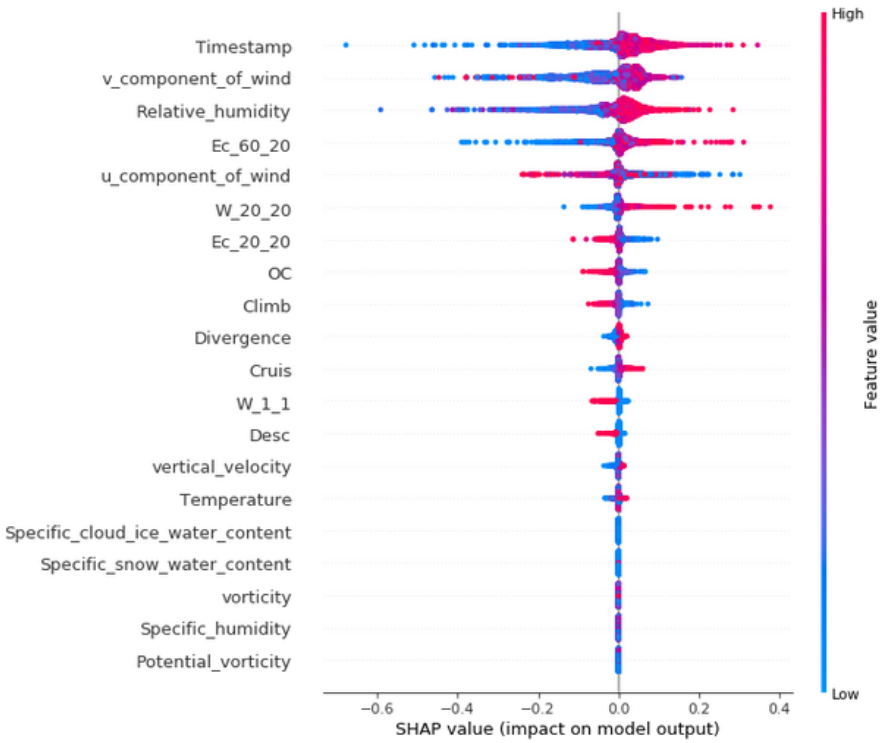
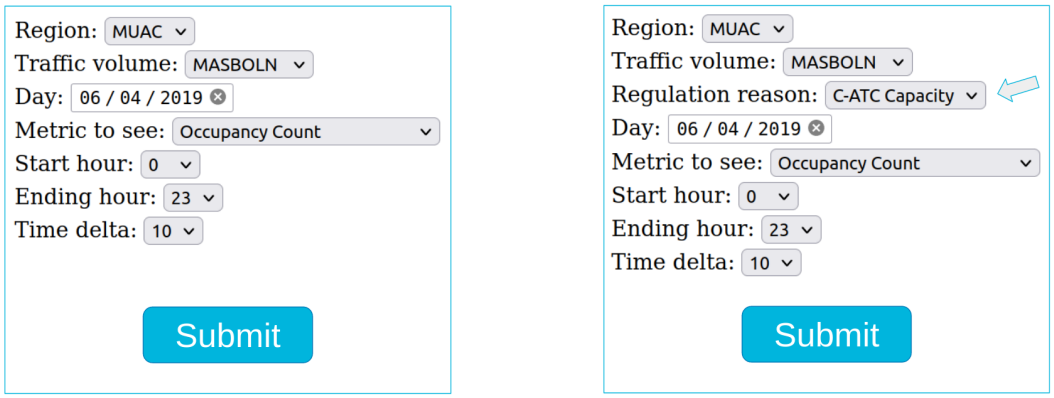



**Figure IV-4:** *SHAP values RNN-based models en-route W-Weather regulations TV B3LL*

## IV.7 Advice capabilities

Because both this and the previous Chapter are related to the same experiment (ATFM regulations at the TV level), the advice generators developed are almost identical. The same web application presented in the previous Chapter (see Section III.7.1) has been used to display the results for the case study that estimates the probability of W-Weather ATFM regulations. The initial form used by the user to specify the study parameters now has an additional element to specify the *regulation reason*. Figure IV-5(a) shows the form from the case study C-ATC Capacity regulations, and Figure IV-5(b) shows the extended version that allows multiple regulation reasons. The visualization of the predictions as a function of the expected open schema remains identical, using the same legend of color to display the uncertainty of the predictions – Green, orange, and red for certain negative, uncertain, and certain positive predictions, respectively.




(**a**) C-ATC Capacity regulations (**b**) Multiple regulations reasons

**Figure IV-5:** *Web application form evolution*

On the other hand, the other proposed advice capability is the integration of the models in R-NEST. At the moment of writing, the models have been converted to C++ because it is the programming language in which the software was developed. However, it still needs the final integration. The reader is referred to Section II.3.3.1 for further details about the integration into the software R-NEST. Notice that the integration process is almost identical to the previously presented models to predict C-ATC Capacity ATFM regulations. The only difference is the need to compute the extra features related to convective weather.

## IV.8 Discussion

In this Chapter, it has been proposed a supervised ML model architecture able to detect specific *time-steps* that must be regulated due to adverse weather conditions. For specific TVs, the model exhibits an accuracy higher than 80%, a recall of around 85%, and a precision of around 80% across six different TVs belonging to two different regions of the European airspace, being REIMS one of the most regulated regions.

On the other hand, and probably more critical for the problem faced, the model shows a high recall, up to 97% in the best case, detecting *intervals* of time which contains a regulation. Therefore, the model shows that with proper training, it should be able to detect almost all the required regulations with high precision. Additionally, comparing the results obtained, which may be too optimistic, with models that do not use down-sampling techniques could enrich the analysis, showing the expected performance in more realistic conditions.

Related to the model explainability analysis, the *confidence-level analysis* indicates that the model is highly sure about the time-steps which must be regulated, and the *SHAP analysis* identifies the wind and the entry count features as key factors.

Similar to the results obtained in the previous Chapter III, the work done in this Chapter suffers from three major operational constraints. First, deploying specialized models for all the TVs could introduce scalability issues. However, they could be overcome by combining the specialized ones for the most regulated TVs and the general one for the other TVs. Second, despite the small dataset available, the presented models can extract the relationship between the historical data and the implemented ATFM regulations in multiple regions, indicating that it could be used across European airspace. Therefore, a more extensive training dataset should improve the current performance. Third, a priory, the ML models are expected to predict only W-Weather ATFM regulations, limiting the system's usability.

Even though the presented model has been developed and studied for the *pre-tactical* phases of the ATFM services, they could be used for the *tactical* or *post operational* phases using the corresponding input data. Furthermore, the presented results are close to the ones obtained when predicting C-ATC Capacity regulations. Therefore, the combination of both models could be a considerable improvement in the current system used to deliver ATFM services, significantly reducing the workload of the FMP and the NM operators who do this task.

Finally, although other types of regulations exist, it is essential to keep in mind that it is paramount to have related input data when using supervised models. For instance, predicting S-Staffing regulations will require access to a data source with information about the available staff or the expected working configuration at each moment.

# V

# Reinforcement Learning for Demand-Capacity Balancing

Traffic growth and changes in traffic patterns have caused increasing congestion and delay in European airspace. The Central Flow Management Unit (CFMU) continually seeks and develops methods to improve traffic flow management to reduce delays and congestion (Tibichte & Dalichampt, 2014). To this end, and taking into account the available literature, as a research question, this Chapter aims to investigate whether a **Reinforcement Learning (RL) techniques** are able to **smooth the traffic of demand-capacity imbalances** without sharing explicit information between agents, and using an approach whose observation states size does not depend on the number of agents.

The Air Traffic Flow Management (ATFM) problem is formalized as a collaborative Multi-Agent Reinforcement Learning (MARL) system where homogeneous agents representing flights aim to decide on their ground delay jointly with the other flights while not having direct information about the preferences of other flights. The specific goal is to smooth the traffic of already identified ATFM regulations in specific Traffic Volumes (TVs) using images as input to the system and to ensure efficient utilization of the airspace. The usage of images allows the system to extract its own features for the problem instead of manually deciding which ones are more representative, the input size is independent of the number of agents, and it provides a fixed size of the states ensuring good scalability. Moreover, the images allow the agents to have indirect information about other flights.

It is proposed to investigate two types of RL algorithms to smooth the demand-capacity imbalances: first, algorithms based on discrete actions; second, algorithms based on continuous actions. In both cases, a homogeneous population of agents is used to ensure their behavior is the same. Furthermore, the agent-based paradigm introduced in this Chapter tries to emulate

the first-planned-first-served basis used in the current ATFM approach (Computer Assisted Slot Allocation (CASA)). Only flights outside the regulated sector are candidates to be agents, ensuring that only flights outside the airspace sector will be delayed.

## V.1 State of the Art

Previous research investigated optimization techniques to find optimal resource utilization. Ivanov *et al.* (2017) presented an optimization algorithm to minimize the propagation of ATFM delays to subsequent flights, Bolić *et al.* (2017) introduced an integer programming model for strategic redistribution of flights to respect nominal sector capacities in short computation times for large-scale, or Ruiz *et al.* (2019) investigated a new technique that could improve airspace capacity usage and reduce ATFM delays by improving the slot allocation process of CASA to avoid wasted capacity (empty slots) at regulated sectors.

On the other hand, several works attempt to study the downstream effects of ATFM regulations and propose resolution techniques. Dalmau (2022) used gradient-boosted decision trees to predict the likelihood of a regulated flight re-routing to mitigate the ATFM delay, and Delgado & Prats (2012) proposed to use speed reduction on air to absorb ATFM delay at no extra cost. Most recent works on the resolution of Demand-Capacity Balancing (DCB) issues focus on the use of RL techniques. For instance, Fernández *et al.* (2017) proved it is possible to both identify and solve DCB problems comparing three RL algorithms for the pre-tactical phase. Similarly, Spatharis *et al.* (2021) is the result of a set of publications where the DCB is formulated as a hierarchical MARL decision problem with different levels of abstraction. However, a critical drawback of this MARLs approaches, in the context of DCB, issues is that a different agent controls each flight, presenting a severe scalability problem, as hundreds or even thousands of different agents would be required to handle the full European Air Traffic Management Network (EATMN).

In response to the previous scalability limitations, Huang & Xu (2021) presented a collaborative Multi-Agent Asynchronous Advantage Actor-Critic (MAA3C) framework with embedded supervised and unsupervised Neural Network (NN), where only flights crossing airspace sectors with already identified demand-capacity issues are regarded as the candidate agents. This approach improves the scalability and generalization of the system, being able to handle a varying number of agents. As an extension of the scalability issues, Kravaris & Vouros (2022) reviews different deep MARL methods examining their ability to scale up to large agent populations. That is from hundreds up to several thousands of agents. The main conclusion drawn with respect to possible scalability issues is the importance of parameter sharing in large agent populations. It is impractical to train thousands of independent networks for each agent or to utilize an approach whose input size would explode as the number of agents and their observations grew.

Similar research has been conducted outside the EATMN region. In the USA network, Tumer & Agogino (2007) developed a MARL system for Air Traffic Management (ATM) integrated with an air traffic flow simulator - FACET. In Crespo *et al.* (2012) was presented a distributed decision support system for tactical ATFM in Brazil, and traffic flow managers experts analyzed the solutions proposed by the system.

## V.2 Problem formulation

En-route ATFM regulations are located at specific airspace TVs (which can be informally defined as a portion of airspace linked to a sector) where a demand-capacity imbalance is detected. Nowadays, the methodology used to identify where ATFM regulations are required is purely human

and does not rely on automation. Air Navigation Service Providers (ANSPs) define two capacities for the sectors which have to be interpreted by the Flow Manager Position (FMP): the sustained capacity and the peak capacity. The sustained capacity indicates the maximum number of flights that can be operated for a particular time window, while the peak capacity indicates the maximum value for a specific instant of time. Close to the day of operation, capacities are defined based on the Occupancy Count (OC), which considers the expected number of flights inside the traffic volume.

It is possible to have multiple demand-capacity imbalances in the network simultaneously. However, the general principle is that a flight subject to several ATFM regulations is given the delay of the most penalizing regulation, *i.e.*the regulation that issues the largest delay.

### V.2.1 Assumptions

In this work, the following assumptions are considered to define the ATFM delay system for specific traffic volumes:

1. The airspace sectors with a demand-capacity imbalance are known (interval of time with overload, location, and capacity), and squares can be used to approximate their shape;
2. Pre-tactical flight plans are available for each flight before any regulation is applied. The flight plans contain the Scheduled Off-Block Time (SOBT) and the route of the flight. Additionally, it is assumed constant speed for each of the segments composing the routes;
3. Flights are assumed to depart at the planned SOBT;
4. There is one type of agent. There are no aircraft with priority;
5. Financial costs on commercial entities resulting from ATFM decisions are negligible;

There is a deviation from traditional state-of-the-art problems by assuming the demand-capacity imbalances are already known for the sector of study. Assumption 1 is required because this work aims to focus purely on resolving the issues of DCB. Only historical data from regulated intervals and sectors are considered. Also related to assumption 1, approximating the sector's shape as squares aims to reduce the implementation complexity in this preliminary study.

Related to assumption 2, constant speed per segment defining the routes is assumed because they only contain information about the starting/ending location and time. By assuming constant speed between the origin and end of the segments, it is possible to interpolate the location of the flights at intermediate timestamps (see Basora *et al.* (2017) and Corrado *et al.* (2020) as other examples of interpolation).

Also related to the flight plans, assumption 3 considers the flight departs at the planned SOBT, *i.e.*the flights do not have assigned departing windows, or there is no uncertainty about the take-off time.

Assumption 4 aspires to create a prototype that is as fair as possible for all the operators. A homogeneous population of agents guarantees that all flights are treated equally. However, using heterogeneous populations of agents in future work could be interesting from an optimization point of view. For instance, two populations could be used to distinguish domestic or international flights or prioritize transit flights to avoid possible downstream effects such as missing connections. Similarly, assumption 5 is used to emphasize that this prototype focuses on the current used Key Performance Indicatorss (KPIs), although they could be extended according to additional requirements if needed.

### V.2.2 Action Variable - Decision variable

The action variable in this problem corresponds to selecting the ground delay that an aircraft will receive due to a demand-capacity imbalance. At each step $\Delta t$, each agent $i \in \mathcal{N}$ has an associated action variable $a_t^i \in \mathcal{A}$, where $a_t^i$ is the ATFM ground delay.

For discrete action algorithms, the action variable can be defined as:

$$a_t^i \in \mathcal{A}, \quad \mathcal{A} \in \{0, 5, 10, 15\} \tag{V.1}$$

While for continuous action algorithms, the action variable can be defined as:

$$a_t^i \in \mathcal{A}, \quad \mathcal{A} \in [0, 15] \tag{V.2}$$

### V.2.3 State Variable

The state vector $s_t^i \in \mathcal{S}$ includes the information that the population of agents $\mathcal{N}$ uses to determine the actions. Each state $s_t^i$ is defined per flight candidate to be an agent and step of the system.

One of the primary challenges associated with MARL is problem representation. The challenge is in defining the problem in such a way that an arbitrary number of agents can be represented without changing the architecture of the Deep Q-Learning (DQN) or Deep Deterministic Policy Gradient (DDPG). To solve this problem, we propose the usage of image-like tensors where each channel in the images encodes a different set of information from the global state. This representation allows us to take advantage of Convolutional Neural Networks (CNNs), which have been shown to work well for image classification tasks (Krizhevsky *et al.*, 2017) and competitive MARL systems based on images (Egorov, 2016).

The image tensor is of size HxWx3 (shown in Figure V-1), where H is the height, W is the width of our two-dimensional images, and three is the number of channels in the image. The channels can be broken down in the following way:

- **Inside channel**: Contains the representation of the regulated flights inside the sector.
- **Outside channel**: Contains information about the flights outside the sector of study, that is, the flights that may be delayed.
- **Self channel**: Contains information about the agent making the decision.

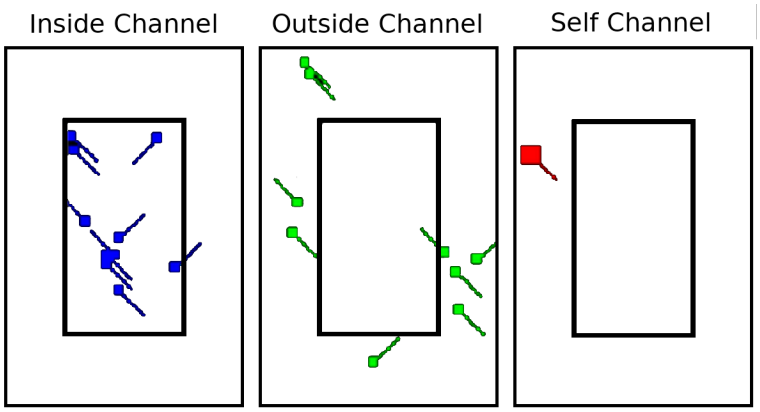



**Figure V-1:** *Three channels image-like representing the input states of the RL system*

Note that the three channels are depicted with white background for clearness, but it encodes zero pixel values. The non-zero pixel values encode the location of the flights, their heading, and the approximate shape of the sector.

### V.2.4 State Transition

The state transition defines a set of conditions that determine how the state $s_t^i \in \mathcal{S}$ evolves along the steps. With every step, the aircraft candidates to be agents decide whether they are going to issue ATFM delay. Three conditions must be verified to ensure a proper transition between states.

The first condition that must be verified is related to the regulations used for the training. Each episode will start using information from a randomly selected historical regulation, and the environment will evolve for a time period $TP$ equal to 60 minutes with a timestep $\Delta t$ equal to one minute. Thus, from the randomly selected regulation, we must guarantee that the regulations will be active for more time than the $TP$. Furthermore, the initial timestamp of the selected regulations is also randomly chosen, ensuring that both the starting moment and the selected regulation are selected randomly in each episode.

The second condition to consider is related to the delay. For each state variable $s_t^i \in \mathcal{S}$, the agent $i$ will produce a new action to cooperatively decide its ground delay to ensure that the demand meets the capacity. Actions equal to zero imply no delay for the flight moving forward on the predefined trajectory. However, if the delay differs from zero, the new delay is added to possible previous delays (cumulative delay).

The last required consideration is related to how the flight is assumed to move forward. A trajectory $T \in \mathcal{T}$ is a time series of segments of the form:

$$T = \{(ID_l, begin_{t_l}, end_{t_l}, lat\_begin_{t_l}, lon\_begin_{t_l}, lat\_end_{t_l}, lon\_end_{t_l})\} \quad l \in [1, m] \tag{V.3}$$

where $ID_l$ is the identifier of the segment, $begin_{t_l}$ the initial timestamp of the segment, $end_{t_l}$ the end timestamp of the segment, $lat\_begin_{t_l}$, $lon\_begin_{t_l}$ the initial latitude and longitude of the segment, $lat\_end_{t_l}$, $lon\_end_{t_l}$ the ending latitude and longitude of the segment, and $l$ is the number of segments used to define the trajectory.

For each of the segments, we assume constant speed. Therefore, the expected velocity of the flight in a particular segment can be defined as follow:

$$v_{ID_l} = \frac{f(lat\_end_{t_l}, lat\_begin_{t_l}, lon\_end_{t_l}, lon\_begin_{t_l}}{end_{t_l} - begin_{t_l}} \tag{V.4}$$

where $f$ is a function that computes the distance between two pairs of coordinates.

Finally, we can compute the aircraft's location at any timestamp, knowing the required segment to use, assuming constant speed in the segments, and considering the imposed ATFM delay.

### V.2.5 Objective Function

Demand reduction is one of the main goals in DCB during the pre-tactical phase. The objective is to smooth the traffic and meet the expected demand with the predefined capacity of the airspace sector. The objective function can be defined with Equation V.5, which corresponds to minimizing the ATFM delay while trying to ensure that the demand meets the sector's capacity for the counting period.

$$\min_{t \in TP} \mathbb{E}\left\{ \sum_{i=0}^{\mathcal{N}} D_i\Big(s_t, \pi^*(s_t)\Big) \right\} \cup V_t \leq C \tag{V.5}$$

where $D_i$ is the ATFM delay of agent $i$, $\mathcal{N}$ is the population of agent, $s_t$ is state of the system at step $t$, $\pi^*$ is the optimal ATFM delay policy, $V_t$ is the OC of the sector, and $C$ is the capacity of the sector.

## V.3 Experimental Setup

This section details the developed DQN and DDPG algorithms, focusing on the dataset used to train the agents, the RL elements, and the parameters of the algorithms.

### V.3.1 Dataset

The data sources required for the development of the case study that uses RL techniques to smooth the traffic of required ATFM are summarized in Table V-1. The Aeronautical Information Regulation and Controls (AIRACs) are used as a source of information about the flight plan intentions of the different flights, and EUROCONTROL data to know which and when TVs were regulated.

**Table V-1:** *Data sources used to smooth en-route C-ATC Capacity ATFM regulations*

| Data source/Format | Period time | Usage | Comment |
|---|---|---|---|
| AIRAC | June, July, August, September 2018 | Flight plans | M1 traffic |
| EUROCONTROL | June, July, August, September 2018 | ATFM regulations | Boolean |

In the EATMN, a wide variety of regulations are applied due to many reasons across different traffic volumes. The study done in this Chapter focuses on **C-ATC Capacity ATFM regulations**, which are those regulations purely related to demand-capacity imbalances. Moreover, because of the huge number of sectors, we have focused our attention on the Maastricht Upper Area Control Centre (MUAC) region. In particular, to the sector *EDYYBOLN* with the associated traffic volume *MASBOLN*. The main reason behind the selection of this particular sector is because it is one of the most regulated airspace regions in the MUAC area, which will guarantee enough variety of samples to train the RL agents. The available dataset contains around 200 C-ATC Capacity ATFM regulations for en-route traffic along 71 different days, with a mean number of regulations per day equal to 1.7 and a mean duration per regulation of 97.08 minutes.

### V.3.2 Reward function

RL algorithms learn from the interactions with an environment, which provides a reward according to how good the agent's action was. The reward function is crucial because different reward structures will result in different system performances.

Previous research has investigated different reward functions. Typically, the literature shows that researchers mainly focused on delay and congestion without considering fairness impact on different commercial entities (Tumer & Agogino, 2007; Agogino & Tumer, 2009). Similarly, Spatharis *et al.* (2018) also took into account the amount of time the agents contributed to the

demand-capacity imbalance. Fairness is usually measured by financial costs imposed on commercial entities resulting from ATFM decisions (Cruciol *et al.*, 2013).

In our case, as a proof of concept using images, we want to focus on delay and congestion. The reward function $G(z)$, written as Equation V.6, consists of three main components: the number of flights delayed $M(z)$, the delay itself $D(z)$, and the demand-capacity ratio $I(z)$:

$$G(z) = -\beta M(z) - \delta D(z) - \lambda I(z) \tag{V.6}$$

where $z$ represent the system under evaluation, $M(z)$ and $D(z)$ represent delay, and $I(z)$ the congestion. $\beta$, $\delta$, and $\lambda$ are the weights used to adjust the income penalty in the evaluation function. Note that the reward function is based on penalties.

In this Chapter, the main goal is to solve DCB issues; therefore, the weight $\lambda$ is set to 5 to penalize the agent when the imbalance is not solved strongly. Then, the second objective is to smooth the demand-capacity imbalances with the minimum delay, using a $\delta$ equal to 2. Finally, because only small delays are allowed, $\beta$ is set to 1 to allow the agent to delay the flight as desired.

When ATFM delay is issued, the number of aircraft entering the airspace traffic volume is reduced; thus, the congestion is relieved. However, this restrictive measure has negative effects on the ATM network. Equation V.7 counts the number of delayed flights and Equation V.8 computes the total delays imposed:

$$M(z) = \Theta(\mathcal{N}) \tag{V.7}$$

where $\mathcal{N}$ is the population of agents, and $\Theta$ is a function that counts the number of flights that received ATFM delay.

$$D(z) = \sum_{t=0}^{P} d_{i \in \mathcal{N}, t} \tag{V.8}$$

where $D$ is the total ground delay, $\mathcal{P}$ is the counting period, and $r_i^t$ is the imposed ground delay at step $t$ for each agent $i \in \mathcal{N}$.

It is required to compute the number of aircraft at the current step to determine the congestion severity in the airspace sector; that is, the excessive number of aircraft in the sector. The congestion function $I(z)$ is given by:

$$I(z) = \begin{cases} (V - C)^{(V-C)} & V > C \\ 0 & Otherwise \end{cases} \tag{V.9}$$

where $V$ is the number of aircraft in the sector (*i.e.*demand), and $C$ is the pre-defined pre-tactical capacity. Note that the function is characterized exponentially with respect to the excessive number of aircraft in a sector.

### V.3.3 Deep Q-Learning

In this work, the first RL algorithm proposed to study to optimize the ATFM delay is DQN following the approach proposed in Mnih *et al.* (2013). It operates directly on RGB images to play Atari games, uses experience replay to store the agents' experiences, and uses a second target network.

At the beginning of each episode, a new initial state is set. Subsequently, for each step and flight candidate to be an agent, an action is chosen either randomly or greedily and stored in the

replay buffer. In the first episode, the $\epsilon$-greedy strategy has an $\epsilon$ equal to 1, forcing agents to explore. However, this value linearly decreases until it reaches 0.01, ensuring the agents prioritize exploitation in the last episodes.

The input to the system is the images used to obtain the agent's experience tuple of the form $(s_t, a_t, r_t, s_{t+1})$, where $s_t$ is the starting image-like state, $a_t$ is the join actions taken, $r_t$ is the reward received, and $s_{t+1}$ is the new state of the system. The replay buffer stores the last 25,000 experience tuples, and batches with 64 samples are randomly selected to train the NN computing the target value and the respective loss. This loss is the minimum squared error of the predicted and target values, and the Adam optimizer (Kingma & Ba, 2015) is used. After training the online network, the weights of the target network are also updated.

The input layer of the NN takes as input the 150x100x3 images. The first layer convolves 32 8x8 filters with stride 4 and uses a Rectified Linear Unit (ReLU) activation function. The second layer is a batch normalization layer (Ioffe & Szegedy, 2015). The third layer convolves 64 4x4 filters with stride 2 using a ReLU activation function. The fourth layer is a batch normalization layer. The fifth layer convolves 64 3x3 filters with stride 1 and uses a ReLU activation function. The sixth layer is a batch normalization layer. The final hidden layers are a fully-connected with 256 rectifier units and a Droupout layer with a rate of 0.5. The output layer is a fully-connected linear layer with a single output for each valid action. The output of the NN corresponds to the predicted Q-values of the individual action for the input state. The main advantage of this type of architecture is the ability to compute Q-values for all possible actions in a given state with only a single forward pass through the network. Table V-2 shows the remaining hyper-parameters.

**Table V-2:** *Hyper-parameters for the Deep Q-learning algorithm*

| Hyper-parameter | Value | Description |
|---|---|---|
| Episode | 1000 | Total number of training episodes |
| Max steps | 60 | Maximum number of steps per episode |
| Number of actions | 4 | Number of different actions |
| Discount factor | 0.99 | Discount factor of future rewards |
| Learning rate | 0.00025 | Learning rate used by the optimizer |
| Initial $\epsilon$ | 1 | Initial value for exploration |
| Final $\epsilon$ | 0.1 | Minimum value for exploration |
| Target update | 4 | Step frequency to update the target network |

### V.3.4 Deep Deterministic Policy Gradient

The second algorithm studied to optimize ATFM delays is DDPG. It is proposed to follow the approach presented in Lillicrap *et al.* (2015), which adapts the ideas underlying the success of DQN to continuous actions. DDPG is an actor-critic method, where a parameterized actor function $\mu(s)$ specifies the current policy by mapping states to actions while the critic $Q(s, a)$ learns how *good* is the action. Similarly to our previous approach, this implementation of DDPG directly learns from raw pixel information, using a replay buffer, and throughout the use of target networks (one for the actor and one for the critic).

The chosen NN for the actor takes as input 150x100x3 images. The first layer convolves 32 8x8 filters with stride 4 and uses a ReLU activation function. The second layer is a batch normalization layer. The third layer convolves 64 4x4 filters with stride 2 and uses a ReLU activation function. The fourth layer is a batch normalization layer. The final hidden layers are a fully-connected with

256 rectifier units and a droupout layer with a rate of 0.5. The output layer is a fully-connected linear layer with a single output unit.

The chosen NN for the critic takes as input 150x100x3 images and the action predicted by the actor. The first layer convolves 32 8x8 filters with stride 4 and uses a Rectified Linear Unit (ReLU) activation function. The second layer is a batch normalization layer. The third layer convolves 64 4x4 filters with stride 2 with a Rectified Linear Unit (ReLU) activation function. The fourth layer is a batch normalization layer. The fifth layer is a fully-connected with 256 rectifier units. The sixth layer is fully-connected with 128 rectifier units and takes as input the concatenation of the output from the fifth layer and the action from the actor. The output layer is a fully-connected linear layer with a single output unit, with a ReLU activation function to ensure the issued delay is bigger than zero..

A major challenge of learning in continuous action spaces is exploration. An advantage of off-policies algorithms such as DDPG is that we can treat the exploration problem independently from the learning algorithm. We constructed an exploration policy $\mu'$ by adding noise sampled from a noise process $\mathcal{J}$ to our actor policy:

$$\mu'(s_t) = \mu(s_t) + \mathcal{J} \tag{V.10}$$

where $\mu'(s_t)$ is the noised policy, $\mu(s_t)$ is the current policy, and $\mathcal{J}$ is the action noise.

In the first published article based on DDPG and raw pixel images, the authors used the stochastic *Ornstein-Uhlenbeck* process (Uhlenbeck & Ornstein, 1930) to generate random values temporally correlated as action noise. However, in the literature, it can also be found implementations using exploratory noise from a *normal distribution*. Although these exploration approaches are proven to work, recent studies claim that *parameter noise* frequently boosts performance (Plappert *et al.*, 2017). Parameter noise adds adaptive noise to the parameters of the NN policy (actor). It injects randomness directly on the weight of the NN, altering the type of actions the agent makes depending on what the agent currently senses. Different layers of the NN have different sensitivities to perturbation, which is why we add *parameter noise* to the last fully connected layers. The performance of the models is evaluated using all three types of noise.

Last but not least, batches of 64 random samples are used from a replay buffer of size 25.000 to train the networks. Online actor and critic networks are trained by computing the target value and respective loss. The loss is the minimum squared error of the predicted and target values. The optimizer used is Adam. The actor and critic target networks are updated using *soft* target updates instead of directly copying the weights. Table V-3 shows the remaining hyper-parameters.

**Table V-3:** *Hyper-parameters for the DDPG algorithm*

| Hyper-parameter | Value | Description |
|---|---|---|
| Episode | 1000 | Total number of training episodes |
| Max steps | 60 | Maximum number of steps per episode |
| Discount factor | 0.99 | Discount factor of future rewards |
| Learning rate actor | 0.001 | Learning rate used by the optimizer |
| Learning rate critic | 0.002 | Learning rate used by the optimizer |
| Initial $\epsilon$ | 1 | Initial value for exploration |
| Final $\epsilon$ | 0.1 | Minimum value for exploration |
| Target update | 4 | Step frequency to update the target network |

## V.4 Performance evaluation

This section presents the results obtained for both DQN and DDPG implementations, learning from raw pixel images to assign ATFM delay. Section V.4.1 summarizes the KPIs selected to evaluate the performance of the models in this experiment. Section V.4.2 shows the results of the different studies RL techniques with different configurations.

### V.4.1 Key performance indicators

A set of KPI are defined to evaluate the quality of the ATFM delay policy:

- the **sum of the rewards** received by all the agents;
- the **sum of ATFM delay** imposed by the agents;
- the total number of **delayed flights**;
- the **sum of times** the agents delayed a flight;
- the **mean OC** of the sector along the episode.

These KPI's are all relevant when evaluating the ATFM plan on a MARL system. One of the most widely used indicators to evaluate the performance of the agents is the sum of rewards earned at the end of each episode. The total delay imposed by the flights is also crucial because it is one of the indicators to minimize. The total number of delayed flights and the number of times the agents applied a delay (number of actions) can be considered KPIs, showing how those delays are distributed among aircraft and the number of micro-adjustments agents make. The OC is key because it dictates situations with severe demand-capacity imbalances.

### V.4.2 Results

To compare the performance between the different implementations, Figure V-2 shows the trend of the different KPIs using a moving window of fifty episodes. Those values have been obtained in all the cases, periodically testing the policy without exploratory noise.

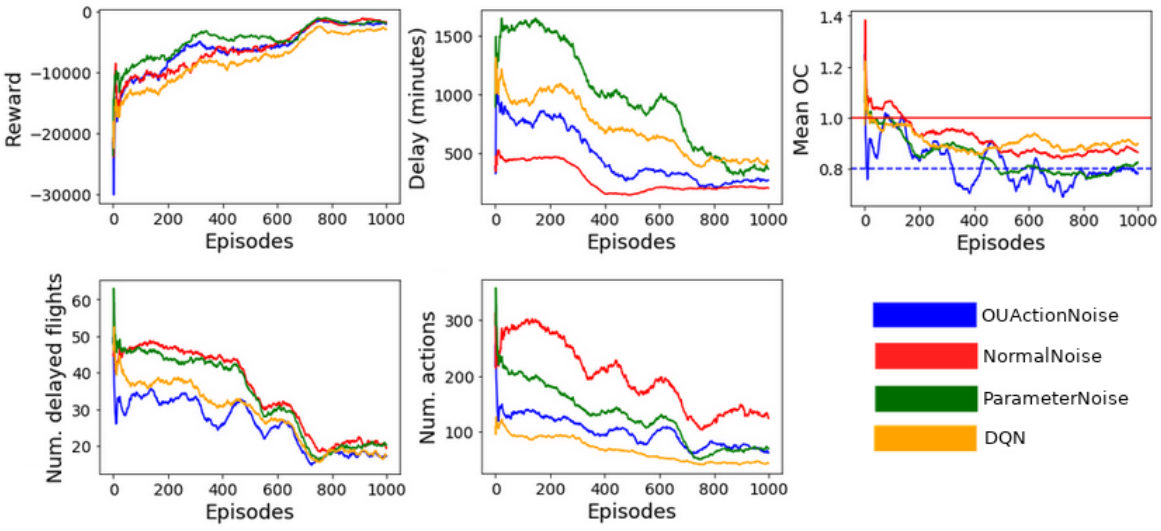



**Figure V-2:** *Trends KPIs used to evaluate the performance of the RL systems*

The results demonstrate the potential of using RL algorithms based on images to solve DCB problems. As expected, the total reward per episode increases with the number of episodes, meaning that the agents are able to improve their policy by gathering experience from the environment. For the last 250 episodes where we can assume convergence of the reward, DQN reported a reward around minus three-thousand while DDPG around one-thousand five-hundred. Note that the reward will always be smaller than zero because the scenarios the agents will see always have DCB issues; thus, ATFM delay is mandatory. From the point of view of maximizing the cumulative reward, DDPG exhibits better performance than DQN.

The total ATFM delay shows a downward trend, denoting that the agents can infer which flights are more efficient to delay. DQN is the algorithm with the largest delay in the last episodes. DDPG with exploratory noise from a *normal distribution* reported the lowest delay while DDPG with *Ornstein-Uhlenbeck* and *parameter noise* reported an intermediate amount of delay. The main reason behind this difference in performance could come from the native characteristics of the algorithms. DQN uses discrete actions that constrain the possible delay values, while DDPG uses continuous actions providing much more flexibility.

The number of delayed flights also decreases with a similar behavior between all the configurations, with an average value of around twenty delayed flights in the last 250 episodes. Although DQN and DDPG with *Ornstein-Uhlenbeck* exploratory noise report slightly better performance, the improvement is minor.

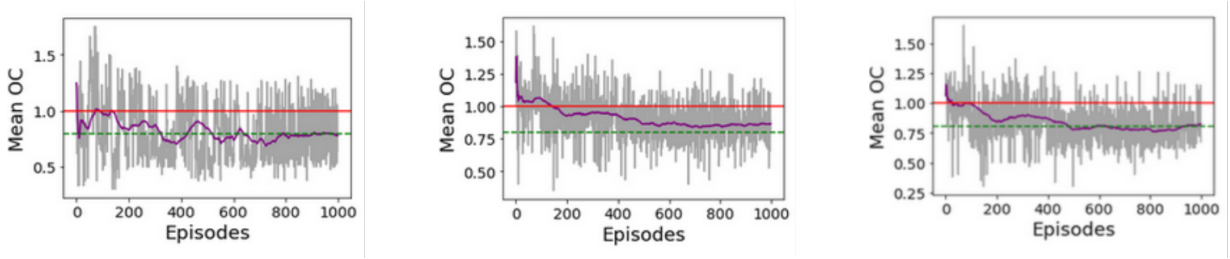
The number of actions applied by the agents shows that DQN is the algorithm with fewer micro-adjustments. DDPG with *Ornstein-Uhlenbeck* and *parameter noise* reported an intermediate similar number of actions. DDPG with noise from a *normal distribution* reported the highest value. This KPI is not directly linked to the goal of solving demand-capacity imbalances. However, it is a good indicator of how many micro-adjustments are required to smooth the expected demand.

Related to the mean congestion of the sector, after six hundred episodes, the sector's mean OC seems to stabilize. DDPG with *Ornstein-Uhlenbeck* and *parameter noise* reports on average an 80% usage of the airspace sector capacity, while DDPG with *normal noise* and DQN exhibits an around 90% usage of the capacity. As a reference, in the European ATM network, the desired occupancy value is around 80% of sector capacity, providing space to absorb unexpected events and ensuring that Air Traffic Controllers (ATCOs) are not overloaded (Niarchakou, 2022).

Looking at the results of the different KPIs, it is not completely clear which approach reports the best overall performance. While DDPG with *normal noise* excels at reducing the overall delay, DQN or DDPG with *Ornstein-Uhlenbeck* achieve a greater reduction in the number of affected flights, and DDPG with *Ornstein-Uhlenbeck* or DDPG with *parameter noise* further optimize the use of sector capacity. To better analyze the behavior of the algorithms from the DCB point of view, that is, focusing on capacity usage to smooth the expected demand, Figure V-3 shows the mean OC per episode for the DDPG implementations. The image shows the collected values per episode (gray) and their trend (purple), the red line represents the sustained capacity, and the green line represents 80% of the sector's capacity.

For the last 250 episodes where the mean OC converges, results from the DDPG with *Ornstein-Uhlenbeck* show the worst performance where many episodes reported a mean OC larger than the sustained capacity. DDPG with noise from a *normal distribution* reports slightly better results with fewer episodes with a demand greater than the sustained capacity on average. DDPG with *parameter noise* reports the best result with the smallest number of episodes with a mean demand larger than the sustained capacity.

Note that even though the algorithms do not keep the mean OC under the sustained capacity for all the episodes, for the last episodes where we can assume convergence in the performance, the mean demand does not exceed the peak capacity. Furthermore, focusing on the *parameter noise* implementation, it can be seen that the frequency and the number of consecutive episodes where




(a) OU (b) Normal (c) Parameterized

**Figure V-3:** *Mean occupancy count per episode for the DDPG implementations.* ***(a)*** *Ornstein-Uhlenbeck noise.* ***(b)*** *Normal distribution noise.* ***(c)*** *Parameter noise*

the demand exceeds the sustained capacity are much smaller than in the other implementations.

Last but not least, we would like to mention that a direct comparison between the results presented in this article and the actual ATFM delay is not feasible since the latter is the result of considering a broader environment. Let us imagine, for example, that a flight crosses two different regulated sectors. Even though the CASA algorithm could impose two different delays, the hypothetical flight would be affected only by the largest one. To directly compare the ATFM delay between the two approaches, a RL model for the two airspace sectors would be needed. Extending the proposed system to a broader region that considers the interaction between neighboring sectors is a relevant point to be studied in future works.

### V.4.3 Case study

This section shows the outcome of the framework for two specific regulations subtracted from the training dataset. The selected regulations are *YBOLN07* from September 7th 2019, and *YBOLN18A* from August 18th 2019. For each of the previous regulations, the RL system based on DDPG and *parameter noise* is used to collect which flights should be delayed and the amount of delay. Then, using this information, the original expected pre-tactical traffic is visualized using the following color schema:

- **Red**: System-suggested flights for regulation
- **Green**: Non-regulated flights outside the sector in the corresponding timestamp
- **Blue**: Non-regulated flights inside the sector in the corresponding timestamp

Figure V-4 shows the results for regulation *YBOLN07* that started at 8:00 am and finished at 10:30 am. As a high-level indicator, the 141 flights crossing the sector linked to regulation *YBOLN07* had a total delay of 556 minutes (delay from *YBOLN07* or any other active regulation); thus, an average delay of 3.94 minutes per flight. On the other hand, our RL system suggests regulating 41 (from the 141 flights crossing the traffic volume) with an average delay per flight equal to 3.71 minutes per flight and a maximum individual delay equal to 21 minutes. Note that the comparison of minutes of delay per flight considers all the regulated traffic crossing the sector independently of the regulation.

Looking at the images, the selected sector (*EDYYBOLN*) has two traffic flows, one from top-left to bottom-right and another from bottom-felt to top-right. Both traffic flows are similarly regulated, indicating that the delay is spread between fights, and the system does not have a preference. However, the RL policy sometimes decides to delay flights that do not completely cross the sector, which seems to be not ideal (see Figure V-4, timestamp 8:57, red fight at the bottom-right).

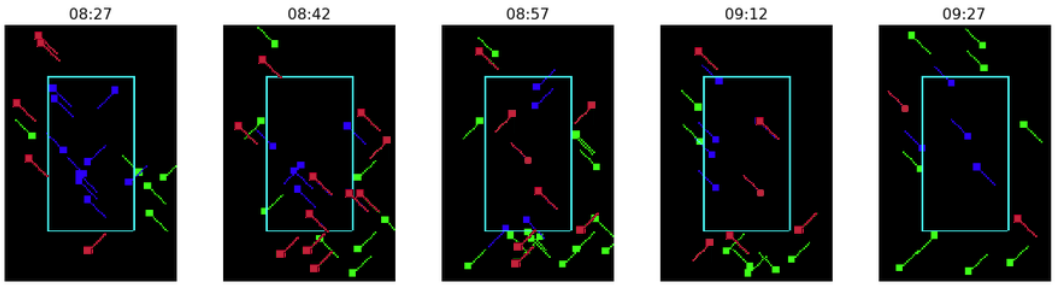



**Figure V-4:** *Advice generator outcome of the RL system for the regulation* YBOLN07

Note that only five images at different timestamps per regulation are shown because of space constraints. Furthermore, there are no regulated flights in the first timestamp to guarantee that agents are not directly penalized due to the demand-capacity imbalance without being able to perform any action.

Figure V-5 shows the results for regulation *YBOLN18A*, which started at 2:00 pm and finished at 4:45 pm. In this case, the flights crossing the sector when the regulation was active received an ATFM delay of 3.39 minutes per flight, while the RL framework regulated 48 (from the 159 flights crossing the traffic volume) with an average delay per flight equal to 3.35 minutes per flight and a maximum individual delay equal to 18 minutes. Notice that, despite the images being more crowded than in the previous case study, the average delay per flight is slightly smaller. 3.94 versus 3.39 for the actual ATFM delay, and 3.71 versus 3.35 using the RL system. This is also the case for the peak delay imposed on individual flights.

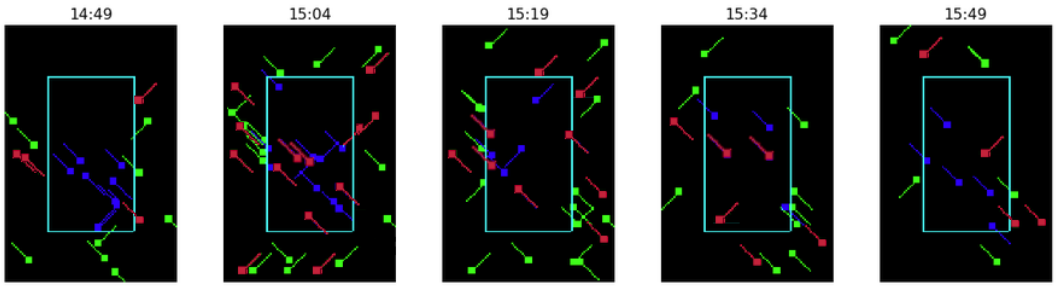



**Figure V-5:** *Advice generator outcome of the RL system for the regulation* YBOLN18A

The results obtained in these two case studies show the potential of the proposed new framework. The RL system is able to solve already detected DCB problems using images with behavior that could be considered valid. However, a deeper analysis is required to obtain further conclusions.

## V.5 Discussion

This Chapter proposes an image-based MARL solution to optimize ATFM delay in the European network. The goal is to maximize the usage of the airspace sector's capacity while minimizing the ground delay. The proposed approach compares DQN and DDPG algorithms with experience replay buffers, target networks, and different strategies for exploration. Although the obtained results did not lead to a clear conclusion about which algorithm configuration best fits the problem, DDPG arises as a promising candidate. It exhibits lower overall ATFM delay and a mean OC closer to optimal values, especially if *parameter noise* is used for exploration during training.

The results obtained as a first step towards devising MARL methods for deciding on ATFM

delay policies using pixel images are promising. The proposed system can successfully solve complex real-world DCB problems. Moreover, the work presented in this Chapter could contribute to improving the usage of the airspace sector's capacity and reducing current delays. One relevant aspect to highlight from this research is that the approach based on images for DCB problems provides a scalable architecture that allows the representation of an arbitrary number of agents without changing the state variables architecture. This characteristic is especially relevant when working on the entire European ATM network, where thousands of flights are operated daily.

Despite the promising results obtained in this proof of concept, some limitations are identified. Related to the delay, only positive delays are valid actions for the agents, but adding the option of removing previously issued delays could enrich the system, providing more flexibility to the agents.

Associated with flight plans used and the assumption related to the take-off time, the approach presented in the Chapter makes the framework more suitable for Air Traffic Control (ATC) tasks than dealing with pre-tactical ATFM regulations. For instance, it could be more convenient for cherry-picking measures during the day of operation (D0). The deviation between the expected and final location of the flights for specific timestamps could introduce an error too big to be considered during pre-tactical DCB issues. However, overcoming the initial assumptions, for instance, introducing a departing window, could make the proposed framework valid for pre-tactical tasks too.

Another limitation of this work is that only individual TVs are studied, but the issued delay required to solve DCB issues in one specific TV could create imbalances between the demand and the capacity in adjacent sectors. Ideally, it would be necessary to solve the DCB issues and the network level, but to be used in combination with the models presented in Chapter III, it was decided to perform a proof of concept at the level of en-route TVs. The combination of this work with the one presented in Chapter III will create an end-to-end system to precisely identify the airspace sectors with demand-capacity imbalances and propose a possible solution to reduce such demand during the pre-tactical phase.

*Obstacles are those frightful things you see when you take your eyes off your goal*

— Henry Ford

# VI

# ATFM regulations at the flight level

Airlines perform their aircraft assignment between 15 and 7 days prior to the day of operations. With this process, specific aircraft frames are allocated to schedules considering operational constraints, defining the different rotations for their flights through the day of operation (D0). In day prior to operations (D-1), the operation plan is drawn to identify potential network issues and prepare pre-tactical preventing measures, such as aircraft tail swapping or crew reassignment. During D0, flight plans will be updated (up to 3 hours prior to departure), and pre-tactical actions implemented, if needed by the duty manager, in order to minimize the propagation of disruption in the network.

During the operational plan definition, airlines submit multiple flight plans, trying to optimize as much as possible the different rotations of flights during D0. However, flights might experience discrepancies between their plan and execution due to many different factors, particularly demand-capacity imbalances in the network leading to Air Traffic Flow Management (ATFM) regulations.

This Chapter studies the usage of **supervised Machine Learning (ML) models** to predict **ATFM regulations issued to specific flights**, aiming to support the tactical planner or the duty manager when optimizing the different rotations for D0. Concretely, because ATFM regulations are particularly complex, this Chapter studies whether it is possible to predict **four different characteristics** of the ATFM regulations that could be of interest to the airlines. The expected prediction horizon is around 24 hours before Scheduled Off-Block Time (SOBT).

## VI.1 State of the Art

Applying machine learning techniques to Air Traffic Management (ATM) is an active area of research. However, as has been seen in the previous Chapters, less work has focused on the study of ATFM regulations and their characteristics—especially the prediction of regulation for specific flights.

At the flight level, previous research presented a comparative analysis of models predicting ATFM delays for specific Origin-Destination (OD) pairs (Gopalakrishnan & Balakrishnan, 2017). Their analysis focused on the USA network and studied three different prediction problems between 2 and 24 hours in the future: classification of OD pair delay (delays above or below a threshold), prediction of OD pair delays, and predictions of airport delay. Similarly, in the previous paper Rebollo & Balakrishnan (2014), the authors used a Random Forest algorithm to predict future departure delays between 2 and 24 hours. In this case, a 19% error was obtained, classifying 100 different OD pairs as above or below 60 minutes.

Another approach used in previous research was to identify similar past days and estimate possible delays through a comparison process. For instance, Gorripaty *et al.* (2016), the authors used a Random Forest algorithm to learn the similarity between days and to infer possible corrective actions that could be applied. They looked at airport demand figures, capacity estimations, and METeorological Aerodrome Report (METAR) data to find the most similar past day to current day-of-operations.

The resilience of the European Air Traffic Management Network (EATMN) was studied in Sanaei *et al.* (2019), focusing on the management of emergent demand-capacity imbalances (tactical phase), regarded as disruptions due to regulations. A more recent approach based on trajectory preferences was presented in De Giovanni *et al.* (2022). The authors studied the potential trade-off between preferences and delays and the potential benefits to the development of the next generation of ATFM tools. Machine learning techniques were used to extract consistent trajectory options in this context.

Last but not least, the other primary source of research related to ATFM regulations at the flight level focused on studying the downstream effects of regulations or possible mitigation actions. Xu *et al.* (2020) proposed a new framework to improve the cost-efficiency for airspace users when facing ATFM regulations. Dalmau (2022) studied the likelihood of airspace user re-routing to mitigate ATFM delay using decision tree models.

Despite the vast research activity on machine learning applications to ATM in the last years, tackling the problem of ATFM identification at the flight level for the pre-tactical phase exhibits a significant gap. Most of the work focused on global delay and downstream effects of issued ATFM regulations.

## VI.2 Problem formulation

ATFM delays are particularly complex. First, when a flight is affected by an ATFM regulation, they are issued with a Calculated Take-Off Time (CTOT) providing a time window for the flight to depart (from 5 minutes prior CTOT to 10 minutes after). If a flight cannot depart within this window, for instance, due to other delays, the ATFM slot will be missed and a new one assigned. This could lead to significant extra delays being issued to the airline as early slots might already not be available. Therefore, CTOTs act as *barriers* in the planning of flights. Suppose the delay is propagated in a way that ATFM slots are missed. In that case, this might have a significant downstream impact even if the initially assigned delay by the regulation is small or even zero. Therefore, airlines need to closely monitor if slots might be missed and notify the Network Man-

ager (NM) as soon as possible to obtain a new CTOT as close as possible to their new Estimated Take-Off Time (ETOT). On the contrary, if the initial delay is large, then some propagation of delay by previous legs can be *absorbed* by the imposed delay due to the ATFM regulation. Even if the flight is ready, it will not be able to depart until its CTOT window.

Second, in some cases, airlines can respond to the ATFM regulations. For example, if the regulation issuing the delay is in the airspace, a new flight plan which avoids the congested airspace, e.g., re-routing or maintaining a lower altitude to avoid entering the airspace, could reduce (or eliminate) the issued delay. Moreover, suppose the aircraft is *ready*, i.e., with the crew and passengers boarded. In that case, messages can be exchanged with the NM to benefit from potential new early slots generated due to delays or cancellations by other flights.

Overall, airlines need to closely monitor flights that have been regulated and actively produce new flight plans and solutions to reduce the impact of this delay on their fleet. As shown, not only if a flight is impacted by ATFM delay, but the characteristics of this (amount of delay and type of regulation) are required as soon as possible for effective fleet management. Therefore, the following questions are considered due to their operational relevance:

- Whether an aircraft is going to be affected by an ATFM regulation: This is the first factor to consider;
- The protected location type of the regulation (airspace or airport): Regulations are divided between those due to issues in the airspace, which might be avoided by modifying the trajectory (e.g., re-routing or flight level capping), and those due to congestion at the arrival airport, which the airline will not be able to avoid;
- If the regulation issues a positive delay or not: In many instances, the CTOT issued to the airline means that there is no delay to be performed with respect to their planned operations. However, this does not mean that the flight is not constrained as it will have to depart within its slots window ([5 minutes before CTOT, 10 minutes after CTOT]). If this constraint cannot be met (e.g., due to reactionary delay), the ATFM slot will be missed, which could lead to a large additional delay. Therefore monitoring these flights is crucial to ensure that delay is not propagated in the network;
- Impact/severity of the ATFM delay if positive: Estimating not only the expected ATFM delay that will be imposed but the uncertainty of this delay.

Therefore to answer the previous questions, **four ML models** are developed to produce individual estimators with different levels of granularity to support the planning process:

1. **Probability ATFM regulations**: Probability of a flight being regulated;
2. **Aerodrome VS Airspace**: For regulated flights, whether the protected regulation location is due to aerodrome or airspace restrictions;
3. **Zero VS Non-zero delay**: If the ATFM delay issued is positive, i.e., non-zero;
4. **Probability distribution ATFM delay**: Expected value and distribution of ATFM delay assigned, if non-zero.

The first two models predict whether a flight is affected by an ATFM regulation and its characteristics for the regulated flights. The latter two models provide an indication of the issued delay. First, it is studied whether the ATFM delay is zero, only reducing the departing slot. Second, for regulated flight with an expected non-zero delay, it is estimated the probability distribution of the possible expected ATFM delay (aka. ground delay).

Because the results of the different models are complementary, the outcome of some models dictates if the other ones shall be used. Figure VI-1 shows the pipeline of the proposed framework.

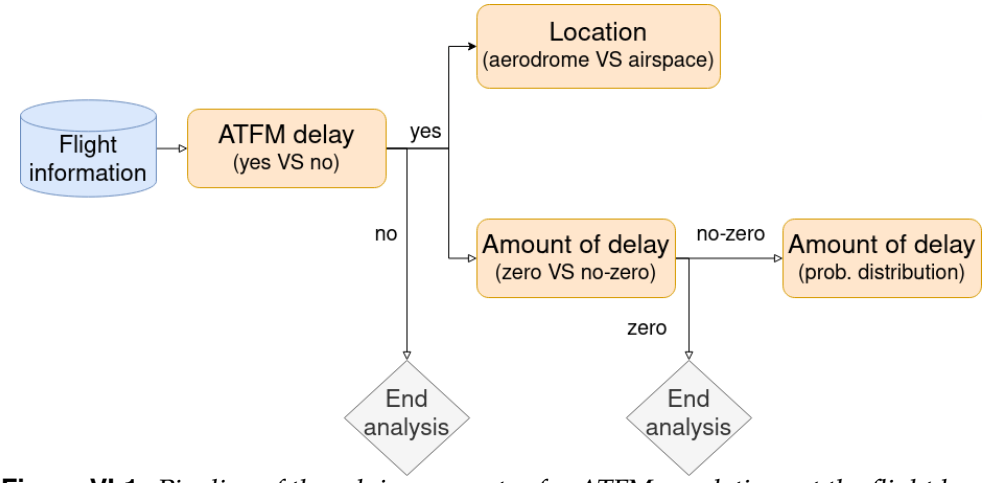



**Figure VI-1:** *Pipeline of the advice generator for ATFM regulations at the flight level.*

### VI.2.1 Assumptions

Two main assumptions are required to predict ATFM characteristics for flight flown by a specific operator. First, it is assumed that the Flight Intentions (FIs) of all flights operated in the same D0 are known. Second, it is assumed that historical traffic information is available to estimate the flight plans of the known FIs.

FIs are assumed to be available to know the expected rotation of the selected operator for D0, as well as the intention of the other operators. It is essential as it is required to estimate the expected demand of both airports and airspace sectors, but also operational information such as the expected departure hour and day.

The second assumption is required to guarantee that the flight plans used to estimate the airport/network demand will be available on D-1, ensuring no data availability issues are introduced in the feature engineering process. Currently, this is the approach followed by EUROCONTROL during the pre-tactical phase, where PREDICT estimates the flight plans when they have not been filled yet (see Section II.4.1.3 for further details about PREDICT).

## VI.3 Data analysis

Data are one of the key aspects of each framework based on using ML models. Section VI.3.1 summarizes the data sources used for this experiment and Section VI.3.2 the Exploratory Data Analysis (EDA) of the labeling for the four ML models.

### VI.3.1 Data sources

The data sources for the prediction of ATFM characteristics at the flight levels are summarized in Table VI-1. ALLFT+ data is used to know the FIs, the static airport characteristics information specifies the size of the departure/arrival airports and whether they are used as a hub, PREDICT as a source of expected flight plans, NOAA is the selected source of weather information, and data from Vueling guarantees that the labeling of the samples is done using information available by the end user. Vueling is the selected operator for this case study due to its volume of operations.

**Table VI-1:** *Data sources used to predict ATFM regulations for individual flights (flight level)*

| Data source / Format | Period time | Usage | Comment |
|---|---|---|---|
| ALLFT+ | 2018 | Features | Flight intentions |
| Airports data | Static | Features | Size and/or hub |
| PREDICT | 2018 | Features | Airport/network demand |
| NOAA | 2018 | Features | Weather |
| Vueling | 2018 | Labelling | ATFM information |

### VI.3.2 Exploratory Data Analysis

The training/testing observations have been labeled according to the information available from the NM in the selected operator. Figure VI-2(a) shows that around 33% of flights were regulated in 2018, and Figure VI-2(b) exhibits that around 80% of the issued regulations were finally applied. As a reference, the available data source contains information for around 201,000 flights.

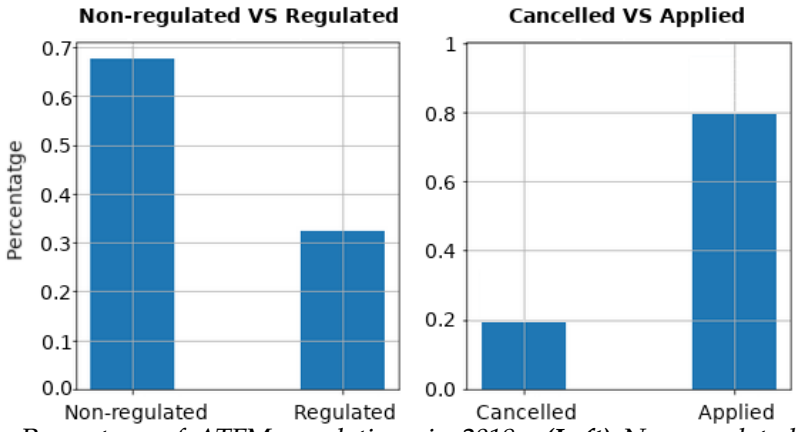



**Figure VI-2:** *Percentage of ATFM regulations in 2018.* ***(Left)*** *Non-regulated VS regulated.* ***(Right)*** *Regulations Cancelled VS Applied*

FigureVI-3 shows that for the active regulations, 61% were due to airspace Demand-Capacity Balancing (DCB) issues while 39% has as a protected location type the aerodrome.

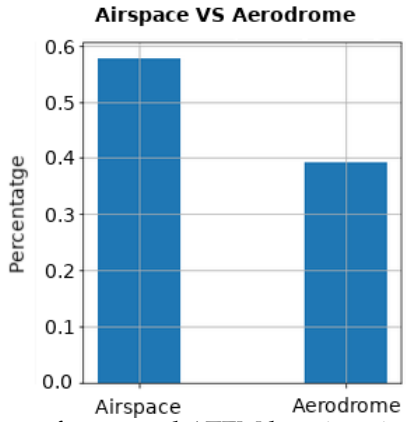



**Figure VI-3:** *Percentage of protected ATFM locations in 2018 for regulated flights*

If the analysis is extended to the actual issued ATFM delay, Figure VI-4 presents the number of occurrences of ATFM delays from zero to one hundred. As can be seen, most of the issued ATFM delays are equal or near zero, concentrating most occurrences under twenty minutes. It is worth mentioning that some outliers over 100 minutes of delay have been identified, but they are not represented in the following image for clarity.

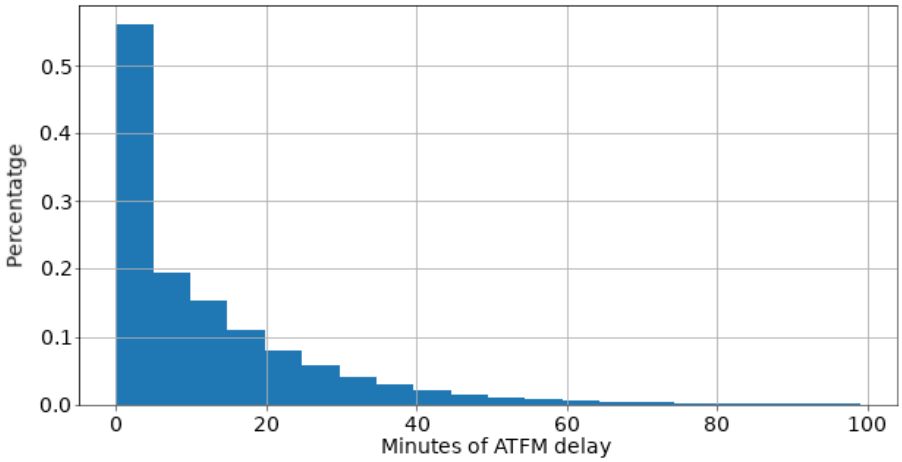



**Figure VI-4:** *Percentage of ATFM delay minutes issued to regulated flights in 2018*

Finally, Figure VI-5 presents the percentage of occurrences of each possible regulation reason, where C-ATC Capacity, W-Weather, and G-Aerodrome are the top three most frequent ones.

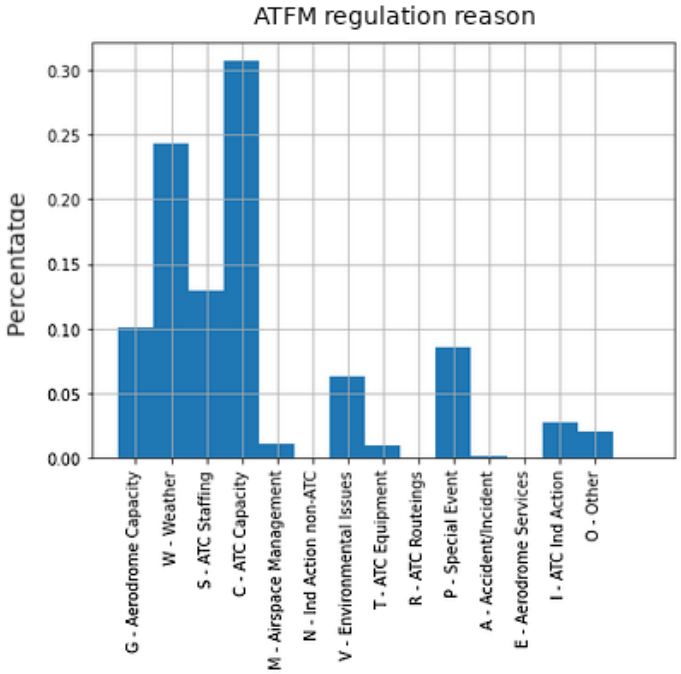



**Figure VI-5:** *Percentage of ATFM regulation reason issued in 2018*

From an operational point of view, the EDA shows the relevance of the selected ATFM characteristics. Moreover, it guarantees a reasonable number of samples to train the different supervised ML models.

## VI.4 Predictive capabilities

As introduced at the beginning of this Chapter, four different ML models are required. However, not all of them belong to the same category, depending on the expected outcome. The probability of ATFM delay, the protected location, and whether the ATFM delay is zero are based on binary classifiers. On the other hand, to estimate the probability distribution of the expected minutes of delay, it is used the combination of a regressor and a multi-output classifier. Table VI-2 summarises the required algorithms for each proposed case study.

**Table VI-2:** *Machine learning model type per ATFM characteristic*

| ML type | ATFM characteristic | Outcome |
|---|---|---|
| Binary classification | Probability ATFM | Prob. none-regulated VS regulated |
| | Protected location | Prob. aerodrome VS airspace |
| | Probability zero delay | Prob. zero VS non-zero delay |
| Regression | ATFM delay | Real value |
| Multi-output classification | ATFM delay prob. dist. | Probability distribution |

The binary classifiers are used as they were designed. The objective is to predict which label is more likely. However, not only the expected amount of ATFM delay but the distribution (and uncertainty) associated with this prediction is relevant to the airline due to the non-linearities of the cost of delay (Cook & Tanner, 2015).

Having introduced the different ATFM characteristics and the required ML models, Section VI.4.1 introduces the input features selected and the output of the different ML models. SectionVI.4.2 shows the feature explainability analysis conducted and presents the obtained results. Section VI.4.3 details the approach followed in selecting the best possible ML algorithms and their hyper-parameters for the binary classifiers. Section VI.4.4 shows the process followed to build a ML system that predicts the probability distribution of expected ATFM delay. Section VI.5.1 collects the evaluation techniques used according to the model type.

### VI.4.1 Inputs and outputs of the models

The objective of supervised ML models is to predict the different target indicators by means of knowing the features. The input features used to predict the different ATFM characteristics at the flight level are particularized below. Section VI.4.1.1 lists all the features computed for all the experiments by their topic. Section VI.4.2 presents the feature explainability analysis based on the correlation between the input features and the target labels.

The are two reasons behind the feature explainability analysis. First, performing this kind of analysis is a good practice when training ML models. Second, to the best knowledge of the author of this thesis, there is no previous work related to feature importance when predicting ATFM characteristics at the flight level. Therefore, this analysis could help determine the information more relevant for the different ATFM characteristics.

#### VI.4.1.1 Input features

The input features selected to predict the different ATFM characteristics are scalar variables that can be grouped into five topics:

- **Operational time**: basic information from ALLFT+ data used as the FIs;
- **Airport static information**: basic static information about the departure/arrival airport per FI. Information from source airport data;
- **Airport demand**: expected number of flights departing from the origin and landing at the destination airports. Input features based on PREDICT data;
- **Network demand**: features linked to the expected congestion in the most crowded Air Traffic Control (ATC) sector the flight is expected to cross. Input features based on PREDICT data;
- **Weather**: numerical indicators related to possible convective weather. Input features extracted from NOAA

For each of the previous topics, Table VI-3 summarizes all the computed features, their data source, and details its definition:

**Table VI-3:** *Input features grouped by topic*

| Topic | Feature | Definition |
|---|---|---|
| Operational time | Hour departure | Hour from SOBT. Value from 0 to 23 |
| | Day week departure | Day week form SOBT. Value from 0 to 6 |
| | Month departure | Month form SOBT. Value from 0 to 11 |
| Airport static information | Size departure airport | Three size {small, medium, large} |
| | Size arrival airport | Three size {small, medium, large} |
| | Hub departure | Used as a hub by the airline {no, yes} |
| | Hub arrival | Used as a hub by the airline {no, yes} |
| Airport demand | Departures hour | Departures respect the same hour |
| | Departures day week | Departures respect the same day of week |
| | Arrivals hour | Arrivals respect the same hour |
| | Arrival day week | Arrivals respect the same day of week |
| Network demand | Normalized OC | OC / avg. OC. Most crowed crossed sector |
| | Normalized OC | OC / max. OC. Most crowed crossed sector |
| | Normalized EC | EC / avg. EC. Most crowed crossed sector |
| | Normalized EC | EC / max. EC. Most crowed crossed sector |
| Weather | Visibility depart/arrival | Directly from NOAA divided by 12000 |
| | Wind depart/arrival | Directly from NOAA (Knots) divided by 30 |
| | u-wind depart/arrival | Directly from NOAA (Knots) divided by 30 |
| | Temperature depart/arrival | Directly from NOAA (F) divided by 125 |
| | Rel. humidity depart/arrival | Directly from NOAA divided by 0.0015 |
| | Geopotential depart/arrival | Directly from NOAA divided by 25000 |
| | Ventilation rate depart/arrival | Directly from NOAA divided by 40000 |

Notice that more input features could be relevant for the different case studies. However, ensuring that the end user (the operator/airline) has access to the required data source to compute them is paramount.

#### VI.4.1.2 Outputs of the models

According to the case study, the expected outcome of the models depends on the selected ML algorithm type. For the binary classifiers that predict the probability of ATFM regulation, the protected location, and whether the issued delay is zero, the outcome of the models is a value between zero and one, showing the probability of the event. For instance, using a threshold equal to 0.5, for the probability of ATFM regulation, a value between 0 and 0.5 indicates the flight will not be delayed. In contrast, a value between 0.5 and 1 predicts that the flight will be regulated.

On the other hand, for the probability distribution of ATFM delay, as the name indicates, we want to predict a probability distribution. First, the regressor is triggered to predict the minutes of ATFM delay. The outcome of the regressor is a real number. Second, the multi-output classifier predicts the probability of each bin used to build the probability distribution of ATFM delay. Third, the outcome of both models is combined, using the outcome of the regressor as an offset for the predicted probability distribution. Figure VI-6 is a simplified visual representation of the approach to predict such a probability distribution.

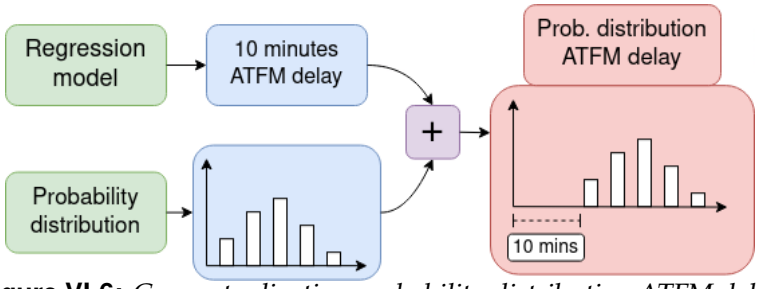



**Figure VI-6:** *Conceptualization probability distribution ATFM delay*

### VI.4.2 Feature selection analysis

Feature selection is the process of identifying and selecting a subset of relevant features from a larger set of features to improve the performance of a ML model. It involves analyzing the importance of each feature and selecting only those that are most informative for the given task. A good example of the effect that the selected features can have on the performance of the models is (Rebollo & Balakrishnan, 2014). Notice that this differs from model explainability, which studies the final impact of a feature in the final trained model.

In this Chapter, a *Filter method* based on an ANalysis Of VAriance (ANOVA) (Judd *et al.*, 2017) between labels and the features for classification and regression tasks are used. The idea is to compute the F-statistic for each feature, which is a measure of the difference between the means of two or more groups of data relative to the variability within the groups. A high F-statistic indicates that the means of the groups are significantly different, and thus the feature is more relevant for the classification task. In regression analysis, the F-test is used to determine whether a set of independent variables (*i.e.*, features) as a whole are statistically significant in explaining the dependent variable (*i.e.*, target variable).

The following images provide the feature selection analysis for the different engineered features for the different ATFM characteristics. The y-axis shows the reported importance of each feature. The logarithm of the F-value is often used in machine learning because the F-value can be very large, and taking the logarithm can make the results more manageable and easier to interpret. In the x-axis, it is presented each of the input features sorted by their importance. Furthermore, the different features are displayed using different colors according to their topic. The legend of the topics is displayed at the top of each image.

Figure VI-7 shows the feature correlation analysis for the ML models that estimate the *probability of ATFM regulation* per flight. The labeling is a boolean variable indicating whether the flight was regulated, and each input feature has been colored according to the corresponding topic presented previously. As can be seen, weather information presents a high correlation with the probability of ATFM. Especially the wind components and speed, geopotential, and temperature. It is also highly correlated with the size of the arrival airport and the network demand features. On the other hand, the operational information and the congestion at the airports are less correlated with the target features, but their correlation cannot be ignored. Finally, the feature that indicates whether the arrival airport is used as a hub by the airline does not add information; thus, it is removed from the final training dataset.

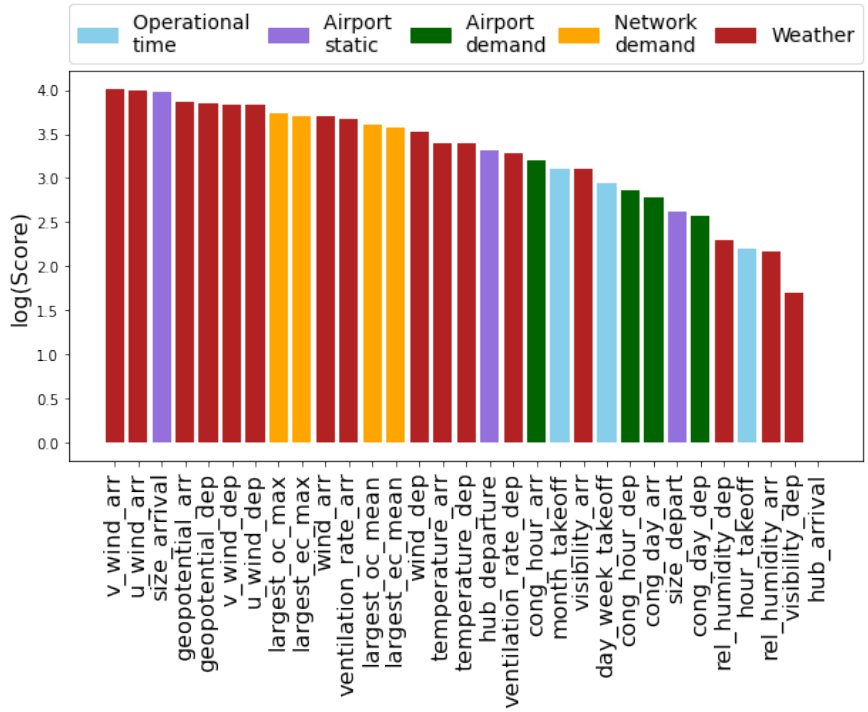



**Figure VI-7:** *Feature explainability for the probability of ATFM regulations*

Figure VI-8 presents the feature explainability analysis for the ML model that estimates the protected ATFM region for regulated flights (aerodrome or airspace). Zero is used to label regulated flights due to aerodrome congestion, while one corresponds to regulated flights with airspace as a protected location region. The correlation analysis shows that the static information about the airports has a high correlation with the labeling, especially the size of the airports and whether the departure airport is used as a hub by the airline. Next, similar to the previous case study, the most correlated weather features are related to the wind, followed by the geopotential and the ventilation rate of the departure airport. The rest of the input features do not present a clear pattern. However, the contribution is not negligible, except for the congestion in the same hour and the ventilation rate of the arrival airport, which are removed from the training dataset as the score is smaller than one.

Figure VI-9 indicates a different pattern of behavior. The static information about the origin/destination airports is highly correlated with the target, but the weather information plays a minor role than in the previous case study. Furthermore, something important to notice is that the overall score of the features is significantly smaller. Previously, the observed scores were around three, while now are around 0.5. It indicates that the overall correlation of the selected features is very low when predicting whether the identified ATFM regulation will issue a delay equal to zero minutes for specific flights. The results make sense as the final imposed ATFM delay comes from

the Computer Assisted Slot Allocation (CASA) algorithm based on the first-planned-first-served principle. For instance, the condition of the network does not determine the issued ground delay. Finally, the congestion at the airports, the month of the year, and the visibility are removed from the final training dataset due to their low correlation.

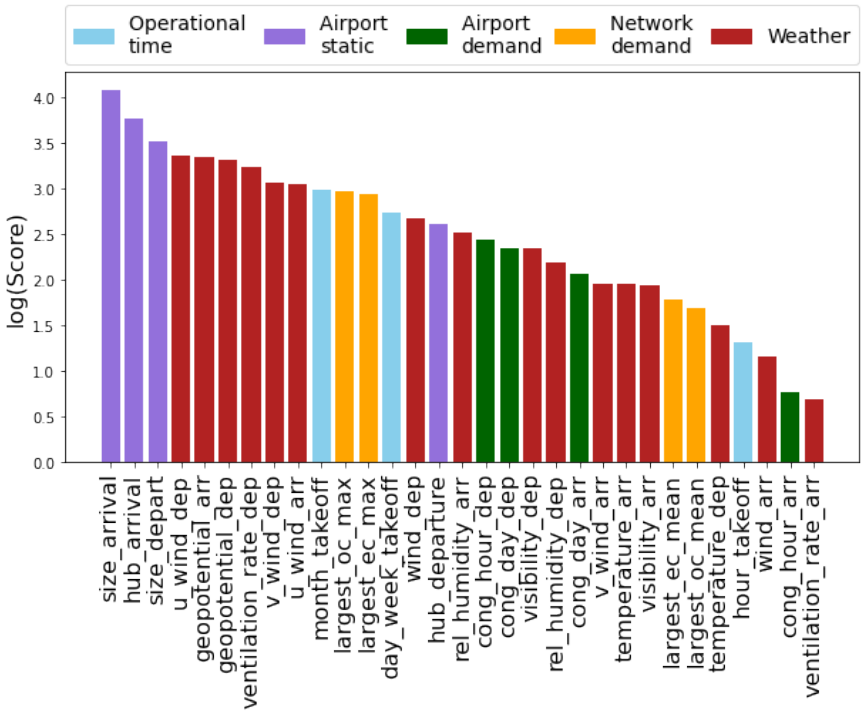



**Figure VI-8:** *Feature explainability for the protected location of ATFM regulations*

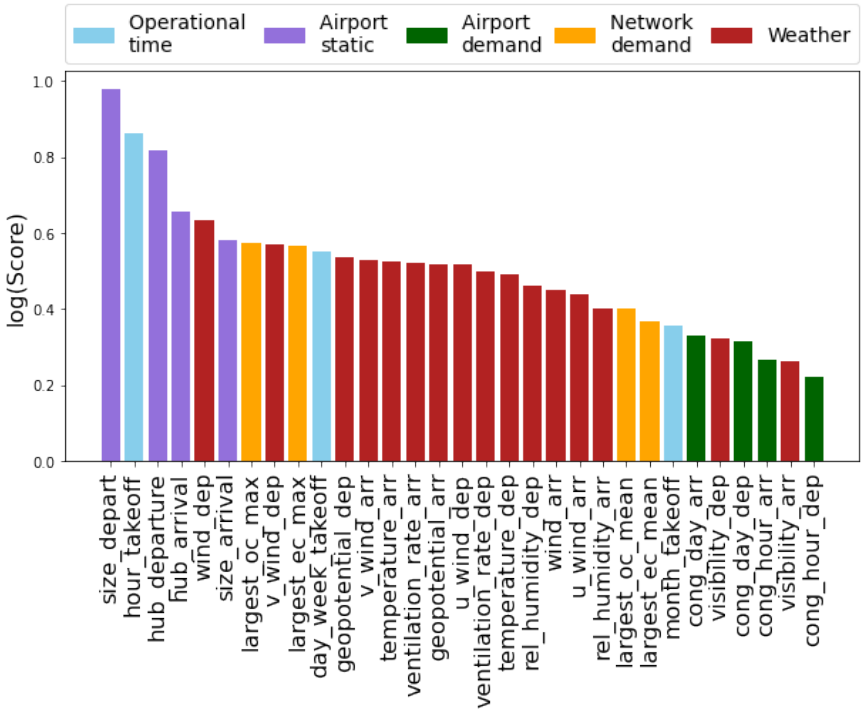



**Figure VI-9:** *Feature explainability for the probability of zero ATFM delay*

Figure VI-10 exhibits the feature correlation analysis between the selected features and the actual ATFM delay. Interestingly, although the selected features were not very correlated with the labeling used to predict whether the delay is zero, this is not the case for the actual ATFM delay with an average score of around two. The most correlated features are the departing hour and the characteristics of the departure airport. Then, the wind, the geopotential, the network congestion normalized by the maximum historical values, and the temperature. Finally, all the features with a score lower than one are removed from the training dataset.

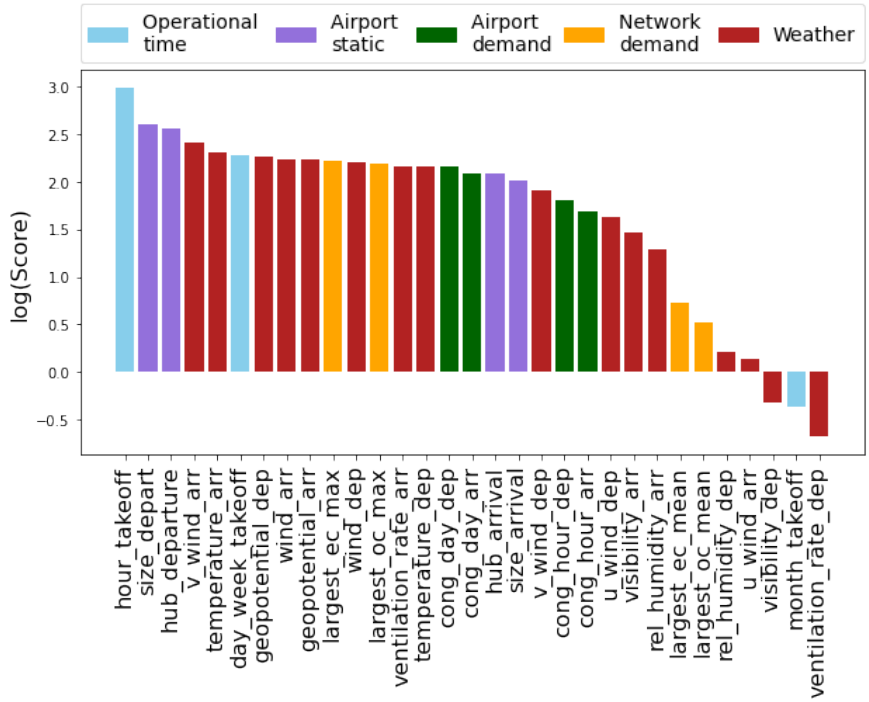



**Figure VI-10:** *Feature explainability for the ATFM delay different than zero*

### VI.4.3 Binary classifiers: probability, location, and zero ATFM delay

After defining the target labels and the input features more relevant for each case study, the first task to perform is the model selection and its hyper-parameters. Hyper-parameters are parameters that are not directly learned within estimators. They are passed as arguments to the constructor of the estimator classes.

There are two main approaches to finding the best possible models and their hyper-parameters: with and without brute force. Brute force exhaustively considers all parameter combinations within the provided search space, while none brute force techniques are based on optimization techniques. In this case study, a brute force technique has been selected due to the small search space required. An initial analysis reported that simple ML algorithms should be enough. Moreover, the approach was validated by experts from the selected operator as the best initial approach.

Concretely, a *GridSearch* analysis (Scikit-learn, 2022b) selects the best ML model algorithm and hyper-parameters, using as a scoring function a *balanced accuracy* (Scikit-learn, 2022a). The balanced accuracy score has been selected for the binary classifiers, ensuring that the selected algorithm and hyper-parameters are as optimal as possible for both classes for the final model. Table VI-4 collects the different studied models and the defined search space. The reader is referred to Scikit-learn (2022c) for further details about the meaning of each hyper-parameter.

**Table VI-4:** *Studied classification algorithms and search space definition*

| ML algorithm | Hyper-parameter | Search space |
|---|---|---|
| MultiLayer Perceptron (MLP) classifier | Hidden layer size | 15, (15, 30), (15, 30, 60), (60, 30, 15), (30, 15), (100, 50, 10) |
| | Batch size | 32, 64, 100 |
| | Solver | lbfgs, sgd, adam |
| | Activation | identity, logistic, tanh, relu |
| | Random state | True, False |
| | Learning rate | 0.001, 0.0001 |
| Random forest classifier | Num. estimators | 50, 100, 150, 200, 250, None |
| | Max. depth | 25, 50, 100 |
| | Criterion | gini, entropy |
| AdaBoost classifier | Num. estimators | 25, 50, 100 |
| | Learning rate | 0.25, 0.5, 0.75, 1 |
| | Algorithm | samme, samme.r |
| Decision tree classifier | Max. features | auto, sqrt, log2 |
| | Max. depth | 25, 50, 100, None |
| | Criterion | gini, entropy |
| | Splitter | best, random |
| Linear SVC | Penalty | l1, l2 |
| | Loss | True, False |
| | Dual | True, False |
| | Class weight | dict, balanced |

### VI.4.4 Probability distribution: ATFM delay

Different approaches for estimating probability distribution can be found in the literature. For instance, quantile regression is one of the most popular because no assumptions are made about the target distribution (Yu *et al.*, 2003). NGBoost enables predictive uncertainty estimation with Gradient Boosting through probabilistic predictions (Duan *et al.*, 2020). Another option could be to train a Artificial Neural Network (ANN), which predicts the parameters of the distribution, among others.

In this thesis, a new approach based on a combination of regression and classification models is used to estimate the amount of ATFM delay and its probabilistic distribution (uncertainty), obtaining the discrete distribution of the possible expected values. The regression model estimates the minutes of ATFM delay, while the multi-output perceptron classification model predicts the error distribution for the previous prediction. The interested reader can refer to De Falco & Delgado (2021). The main reason for using this approach is to test the possibilities this technique can offer and due to consensus among all the partners in Dispacther3, where the study conducted in this Chapter was performed. Figure VI-11 depicts the approach used and the interaction between the different required algorithms.

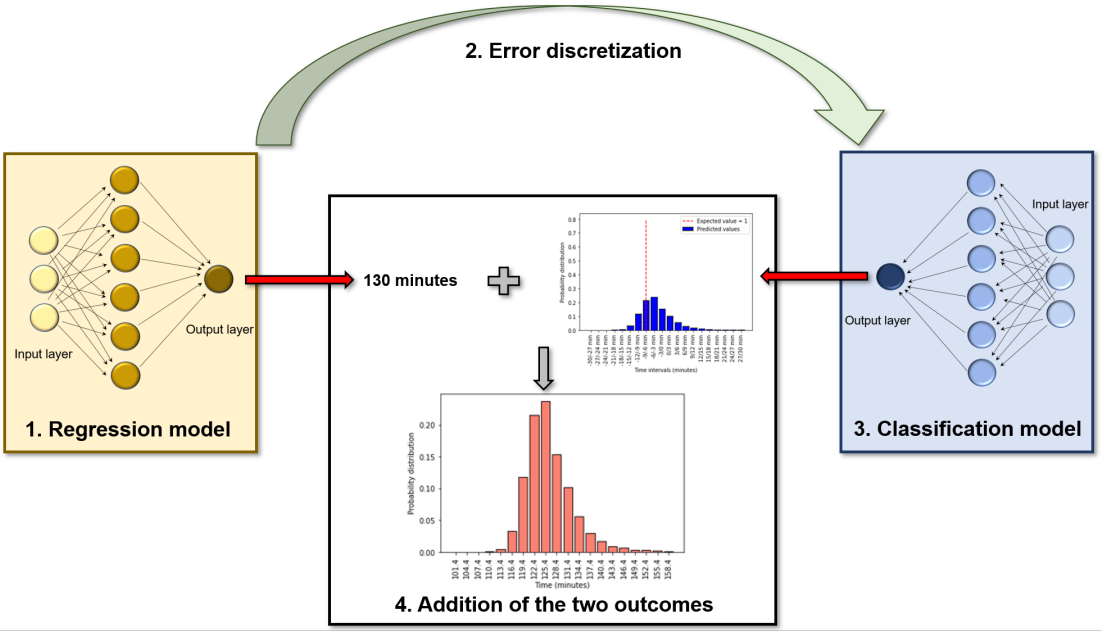



**Figure VI-11:** *Architecture probability distribution ATFM delay.*
*Source: De Falco & Delgado (2021)*

As mentioned previously, the labeling for the regressor comes from the actual ATFM delay imposed on each flight. However, the missing component is the exact labeling for the multi-output classifier that estimated the probability distribution. The goal of the classifier is to predict the probability distribution of the error. Therefore, for the classifier, the labeling is based on computing the difference between the predicted value by the regressor and the actual delay. However, the range of the probability distribution is limited to the most frequent range. Otherwise, the range will be conditioned by the biggest mistake made by the regressor. For example, if the regression model produces most of the predictions with an error between -20 and 20 minutes, this will be the range of values the classifier will try to estimate. Thus, using twenty bins in the distributions, each bin corresponds to a two-minutes error. Figure VI-12 is an example of the labeling for the classifier if the error between the actual ATFM delay and the prediction by the regressor is four minutes.

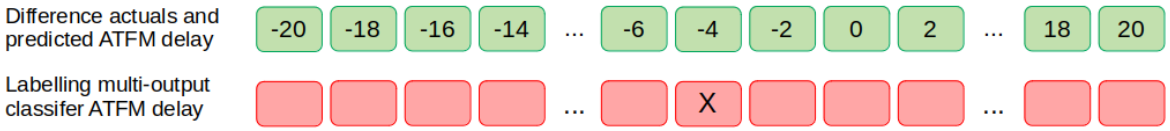



**Figure VI-12:** *Labeling ATFM delay classifier. X indicates elements equal to one.*

Similar to the previous binary classifiers, after defining the target labels, it is time to select the ML algorithms and their hyper-parameters. For consistency, an equivalent *GridSearch* analysis has been conducted to identify the best candidates. Table VI-5 presents the studied ML algorithms and the hyper-parameters search space the regressor that estimates the amount of ATFM delay and Table VI-6 the search space for multi-output classification models that estimate the probability distribution error. In both cases, the performance score is computed using the Mean Absolute Error (MAE) to minimize the deviation between the predicted and actual ATFM delay.

**Table VI-5:** *Studied regression algorithms and search space definition*

| ML algorithm | Hyper-parameter | Search space |
|---|---|---|
| MLP regressor | Hidden layer size | {15, (15, 30), (15, 30, 60), (60, 30, 15), (30, 15), (100, 50, 10)} |
| | Batch size | 32, 64, 100 |
| | Solver | lbfgs, sgd, adam |
| | Activation | identity, logistic, tanh, relu |
| | Random state | True, False |
| | Learning rate | 0.001, 0.0001 |
| Random forest regressor | Num. estimators | 25, 50, 100, 150, None |
| | Max. depth | 25, 50, 100, None |
| | Max. features | auto, sqrt, log2, None |
| | Criterion | squared_error, absolute_error, friedman_mse, poisson |
| AdaBoost regressor | Num. estimators | 25, 50, 100 |
| | Learning rate | 0.25, 0.5, 0.75, 1 |
| | Loss | linear, square, exponential |
| Decision tree regressor | Max. features | auto, sqrt, log2 |
| | Max. depth | 25, 50, 100, None |
| | Criterion | squared_error, friedman_mse, absolute_error, poisson |
| | Splitter | best, random |
| Ridge | Solver | auto, svd, cholesky, lsqr, sparse_cg, sag, saga, lbfgs |

**Table VI-6:** *Studied multi-output classification algorithms and search space definition*

| ML algorithm | Hyper-parameter | Search space |
|---|---|---|
| Multi-output MLP classifier | Hidden layer size | 15, (15, 30), (15, 30, 60), (60, 30, 15), (30, 15), (100, 50, 10) |
| | Batch size | 32, 64, 100 |
| | Solver | lbfgs, sgd, adam |
| | Activation | identity, logistic, tanh, relu |
| | Random state | True, False |
| | Learning rate | 0.001, 0.0001 |

## VI.5 Performance evaluation

This section presents the results obtained predicting all ATFM regulation reasons at the flight level for a specific European airline, Vueling. Section VI.5.1 summarizes the selected evaluation metrics. Section VI.5.2 shows the results predicting the probability of ATFM regulation. Section VI.5.3 the performance of the model that estimates the protected region for regulated flights. Section VI.5.4 the results obtained predicting whether the ATFM delay will be zero. Section VI.5.5 exhibits the performance of the models that estimate the final ATFM delay for regulated flights.

The results focus on a specific operator because it is the expected end-user, being the selected airline one of the operators with a larger volume of flights currently. Therefore, the results obtained could be extended to other airlines. Moreover, Europe is a relatively constrained environment, and other operators should behave similarly.

Before looking into the results, it is worth mentioning that for each model, it is presented the results from the GridSearch analysis to identify the ML algorithm that best fits each problem and its hyper-parameters, the accuracy, recall, precision, and F1-score, but also, the results from the model explainability analysis conducted using SHapley Additive exPlanations (SHAP).

### VI.5.1 Evaluation metrics

The performance of the binary classifier is based on using the accuracy, recall, precision, and F1-score (see Section II.5.1.1 for further details). These metrics will provide a complete overview of the models' performance when predicting the different ATFM characteristics. The accuracy, recall, and precision aim to quantify how good the models are identifying regulated flights, the protected location is airspace, and whether the expected ATFM will be non-zero. However, the opposite target labels are also essential for an operator/airline. In this case, the F1-Score provides an overview of the models' performance predicting both categories. Notice that many metrics could be computed to study the performance of the different models. However, because we decided to train/test models using almost balanced datasets, because of the number of models, and for consistency across Chapters, it has been decided to use simple and well-known metrics. Additional metrics that could be interesting are the logloss, the ROC AUC, or the average precision.

On the other hand, to evaluate the performance of the model that predicts the ATFM delay and the probability distribution, the evaluation process aims to estimate the deviation and uncertainty of the predictions. First, the MAE is used to quantify the deviation between the actual delay and the expected value from the distribution. Second, the uncertainty in the predictions is quantified by computing the average time necessary to cover 90% of the probabilities. Third, the number of hits quantifies how many times the actual ATFM delay is within the predicted distribution. The less uncertain, the narrower the probability distribution uncertain because fewer minutes are likely to be the final ATFM delay. Figure VI-13 is an example of the expected outcome and the required elements to evaluate the performance. In red, the actual delay is represented, and the blue bars are the predicted probability distribution.

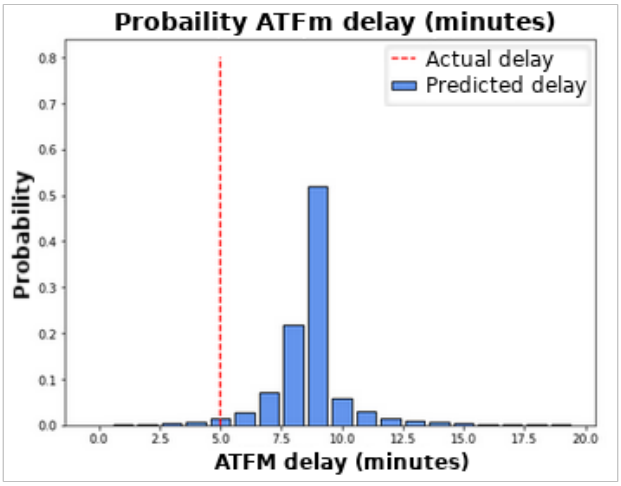



**Figure VI-13:** *Example actual and probability distribution ATFM delay*

### VI.5.2 Probability ATFM regulations

Table VI-7 collects the results from the GridSearch analysis based on a balanced accuracy. The GridSearch analysis reports that the ML algorithm with the best-balanced accuracy is a **Random forest classifier** using a criterion equal to 'gine', a maximum depth of 50, and 200 estimators.

**Table VI-7:** *GridSearch analysis probability ATFM regulations*

| ATFM characteristic | ML algorithm | Balances accuracy score |
|---|---|---|
| Probability ATFM (yes VS no) | MLP classifier | 0.63 |
| | **Random forest classifier** | **0.81** |
| | AdaBoost | 0.7 |
| | Decision tree classifier | 0.7 |
| | Linear SVC | 0.68 |

Table VI-8 exhibits the accuracy, recall, precision, and F1-score obtained. As can be seen, the model can correctly predict most of the regulations with an accuracy of 0.82. Furthermore, it can properly identify non-regulated and regulated flights, reporting an F1-score equal to 0.82. Figure VI-14 shows the confusion matrix obtained predicting ATFMs regulations using a Random forest classifier, 172,111 and 41,692 observations for training and testing respectably.

**Table VI-8:** *Accuracy, recall, precision, F1-Score probability ATFM regulations*

| Accuracy | Recall | Precision | F1-score |
|---|---|---|---|
| 0.82 | 0.81 | 0.82 | 0.82 |

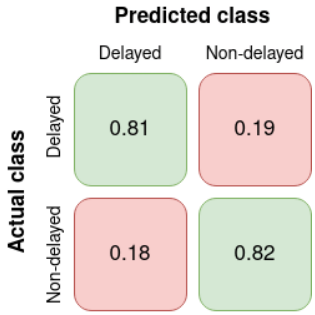



**Figure VI-14:** *Confusion matrix probability ATFM regulations*

Figure VI-15 depicts the SHAP analysis of the trained model to understand better its behavior and the impact of the different input features.

The model prioritizes the size of the arrival airport, the expected wind, the network demand, the geopotential, and operational information such as the day of the week or the hour. Aerodrome ATFM regulations are issued due to high demand at the destination airport; thus, it makes sense to take into account the size of the arrival airport because a bigger airport implies more traffic and higher probabilities of regulations. The relevance of the wind at the arrival airport can drive similar conclusions. The expected network demand is directly related to airspace ATFM regulations, and the geopotential is linked to the altitude of the airports, e.g., to the likelihood of adverse

header conditions. Finally, the day of the week and the expected departure hour are highly relevant because most of the regulations are issued in the morning to avoid downstream effects or on the weekends due to the larger volume of traffic.

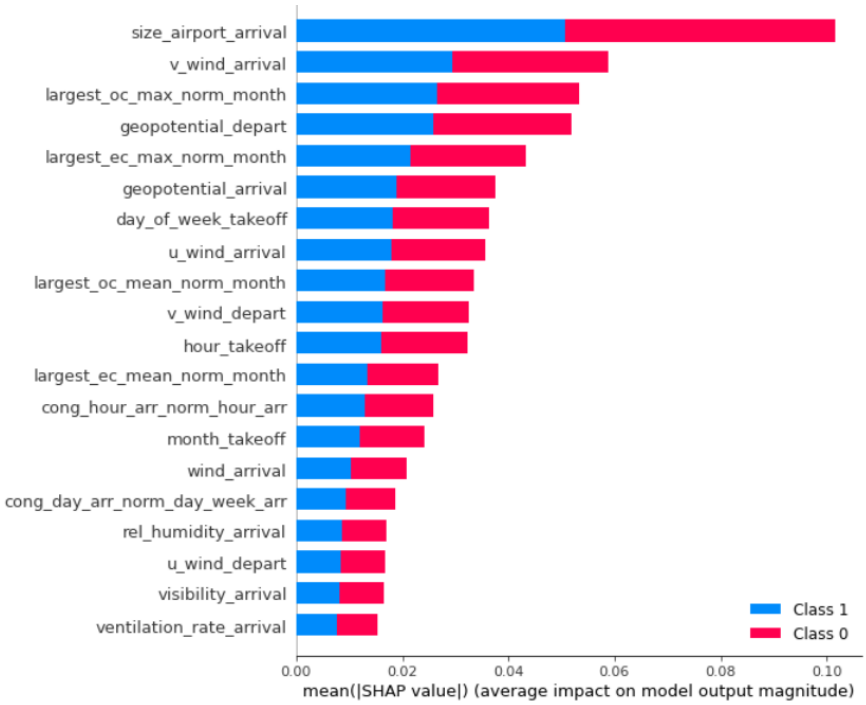



**Figure VI-15:** *SHAP analysis probability ATFM regulations*

### VI.5.3 Location ATFM regulation

Table VI-9 shows the results from the GridSearch analysis based on a balanced accuracy. The GridSearch analysis indicated that the ML algorithm with the best-balanced accuracy is a **Random forest classifier** using a criterion equal to 'gine', a maximum depth of 50, and 200 estimators.

**Table VI-9:** *GridSearch analysis protected ATFM location*

| ATFM characteristic | ML algorithm | Balances accuracy score |
|---|---|---|
| Protected location region (aerodrome VS airspace) | MLP classifier | 0.71 |
| | **Random forest classifier** | **0.86** |
| | AdaBoost | 0.76 |
| | Decision tree classifier | 0.78 |
| | Linear SVC | 0.71 |

Table VI-10 exhibits the accuracy, recall, precision, and F1-score obtained. Similar to the previous case study, the model can correctly predict whether the protected location was aerodrome or airspace, reporting an F1-score equal to 0.86. However, the confusion matrix reveals that the model predicts aerodrome ATFM regulations more accurately.

**Table VI-10:** *Accuracy, recall, precision, F1-Score protected ATFM location*

| Accuracy | Recall | Precision | F1-score |
|---|---|---|---|
| 0.87 | 0.84 | 0.89 | 0.86 |

Figure VI-16 shows the confusion matrix obtained predicting the protected location of issued ATFMs regulations using a Random forest classifier, 56,146 samples for training, and 14,037 samples for testing.

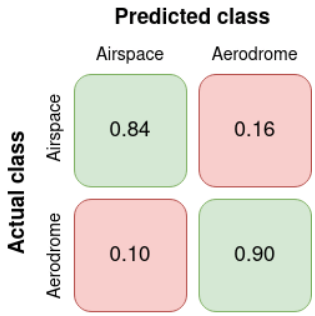



**Figure VI-16:** *Confusion matrix protected ATFM location*

Figure VI-17 presents the SHAP analysis of the trained model. Clearly, the size of the arrival airport is the most relevant input feature for the model. Then, the most relevant features are whether the arrival airport is used as a hub by the airline, some operational information such as the day of the week or the departing hour, and weather information mainly related to the expected wind. All these features are mainly related to aerodrome regulations, so it makes sense that the model performs better when predicting this class.

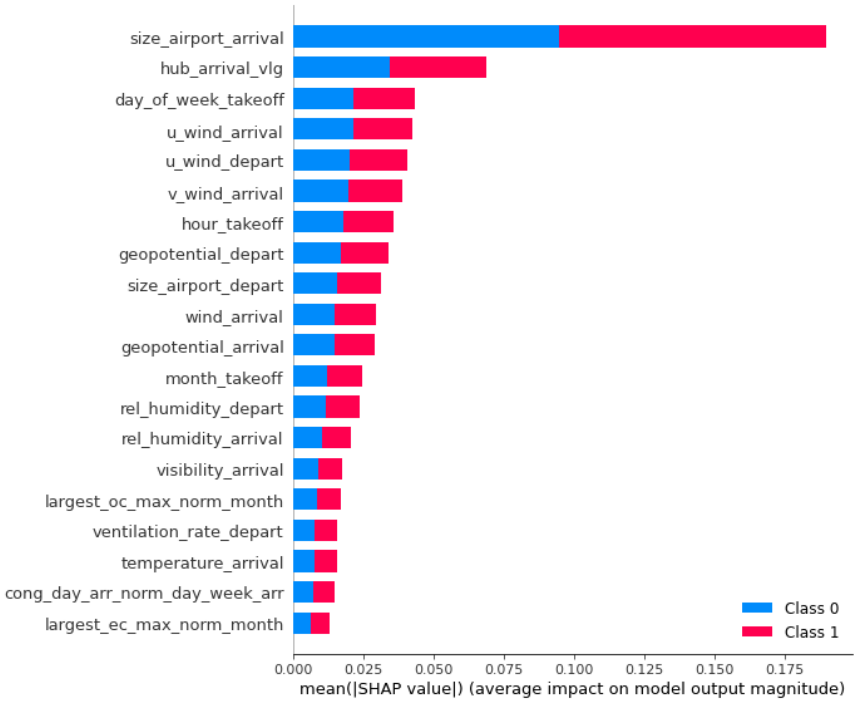



**Figure VI-17:** *SHAP analysis protected ATFM location*

### VI.5.4 Zero VS Non-Zero delay

Table VI-11 summarizes the results from the GridSearch analysis based on a balanced accuracy, indicating that the ML algorithm with the best-balanced accuracy is a **Random forest classifier** using a criterion equal to 'gine', a maximum depth of 50, and 25 estimators.

**Table VI-11:** *GridSearch analysis probability zero minutes ATFM delay*

| ATFM characteristic | ML algorithm | Balances accuracy score |
|---|---|---|
| Probability zeros delay (zero VS non-zero) | MLP classifier | 0.60 |
| | **Random forest classifier** | **0.71** |
| | AdaBoost | 0.55 |
| | Decision tree classifier | 0.6 |
| | Linear SVC | 0.51 |

Table VI-12 shows the accuracy, recall, precision, and F1-score obtained. As can be seen, this is the most challenging problem from a ML perspective. We have seen this during the features correlation analysis, which can be seen in the performance evaluation. The random forest exhibits an accuracy of 0.68 and an F1-score of 0.69, the binary classifier with the worst performance. Suppose we focus our attention on the confusion matrix. In that case, the model tends the overestimate the expected ATFM delay, predicting most of the time that the ATFM delay is going to be different from zero. As mentioned previously, this is the most challenging problem considering the engineered input features. Moreover, it is challenging to distinguish very small delays from delays equal to zero. Figure VI-18 displays the confusion matrix obtained predicting whether the ATFM delay is going to be zero or non-zero using a Random forest classifier, 56,146 samples for training, and 14,037 samples for testing.

**Table VI-12:** *Accuracy, recall, precision, F1-Score zero minutes ATFM delay*

| Accuracy | Recall | Precision | F1-score |
|---|---|---|---|
| 0.69 | 0.67 | 0.69 | 0.69 |

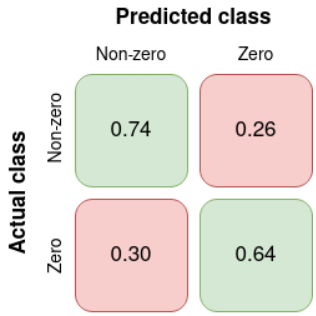



**Figure VI-18:** *Confusion matrix zero minutes ATFM delay*

Figure VI-19 presents the SHAP analysis of the trained model to predict whether the expected ATFM delay of a regulated will be zero or non-zero. Operational information, the geopotential, and the wind are the most relevant features. However, if we focus our attention on the values on the x-axis, the range of reported SHAP values is much smaller than in the previous models

(previously from 0 to 0.1, and now from 0 to 0.05). This also indicates that the model cannot extract a considerable amount of information from the features, which directly relates to the observed low performance.

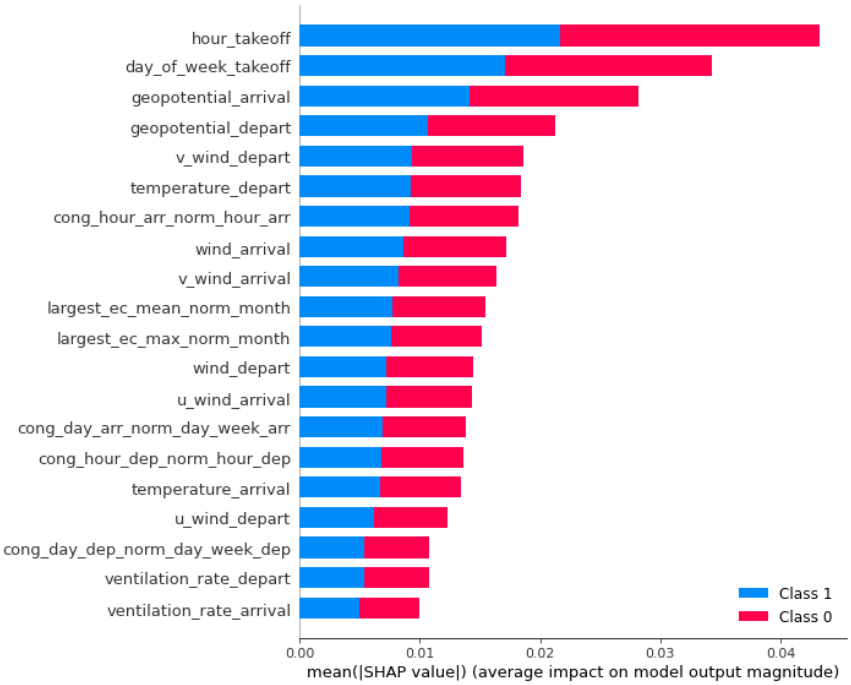



**Figure VI-19:** *SHAP analysis zero minutes ATFM delay*

### VI.5.5 ATFM delay distribution

Table VI-13 collects the MAE from the GridSearch analysis for both the regressor and multi-output classifier that estimate the probability distribution of ATFM delay. The ML algorithm that best estimates the minutes of ATFM is a **Random forest regressor** using a 'squared_error' as the criterion, a maximum depth of 100, a maximum number of features equal to 'auto' which means that all the features are used, and 25 estimators. On the other hand, the best-found multi-output classifier is obtained by using Adam as an optimizer, one hidden layer with 35 neurons and a 'relu' activation function, a random state, and a learning rate equal to 0.0001. Notice that the GridSearch analysis is used to define the input/hidden layer, not the output layer.

**Table VI-13:** *GridSearch analysis ATFM delay*

| ATFM characteristic | ML algorithm | MAE |
|---|---|---|
| Minutes ATFM delay | MLP regression | 25 |
| | **Random forest regression** | **10** |
| | AdaBoost | 50 |
| | Decision tree regression | 14 |
| | Linear SVC | 17 |
| ATFM delay distribution | MLP classifier | 59 |

Table VI-14 shows the MAE and the mean duration in minutes required to cover 90% of the probability (uncertainty), using the previously introduced Random forest regressor and multi-output MLP. The average deviation between the actual ATFM delay and the expected value from the probability distribution is around 9 minutes, with an uncertainty of approximately 12 minutes. The results have been obtained using 56,146 samples for training and 14,037 samples for testing.

**Table VI-14:** *MAE, uncertainty, and number of hits ATFM delay*

| MAE (mins.) | Mean 90% probability (mins.) | Hits (%) |
|---|---|---|
| 9.58 | 12.87 | 0.88 |

Figure VI-20 presents the SHAP analysis of the trained random forest regressor, which estimates the minutes of ATFM delay. In this case, the analysis also presents the hour of departure as the most relevant feature. Next, the most relevant features are the wind, the network demand, the congestion at the airports, and the geopotential. Notice that the results from the departure hour indicate that most non-zero ATFM regulations are implemented in the morning (blue dots). The wind, network demand, and geopotential present the opposite pattern of behavior where larger values are reporting larger SHAP values.

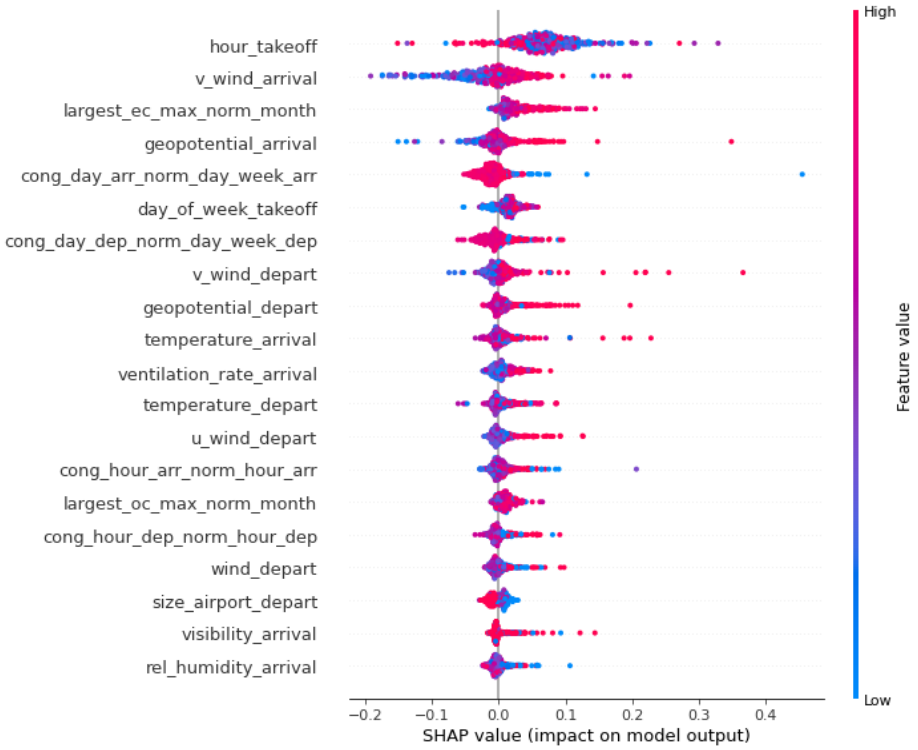



**Figure VI-20:** *SHAP analysis regressor ATFM delay*

Similarly, Figure VI-21 shows the results from the SHAP analysis for the classifier. The most important features are the hour, the day, and the size of the arrival airport. The congestion at the airport and the size of the departure aerodrome follow them. Notice that the most likely classes are in the range [6, 9], which indicates that the classifier is trying to compensate for the possible overestimation of the delay from the regressor. Class zero corresponds to no deviation between the actual and the expected delay.

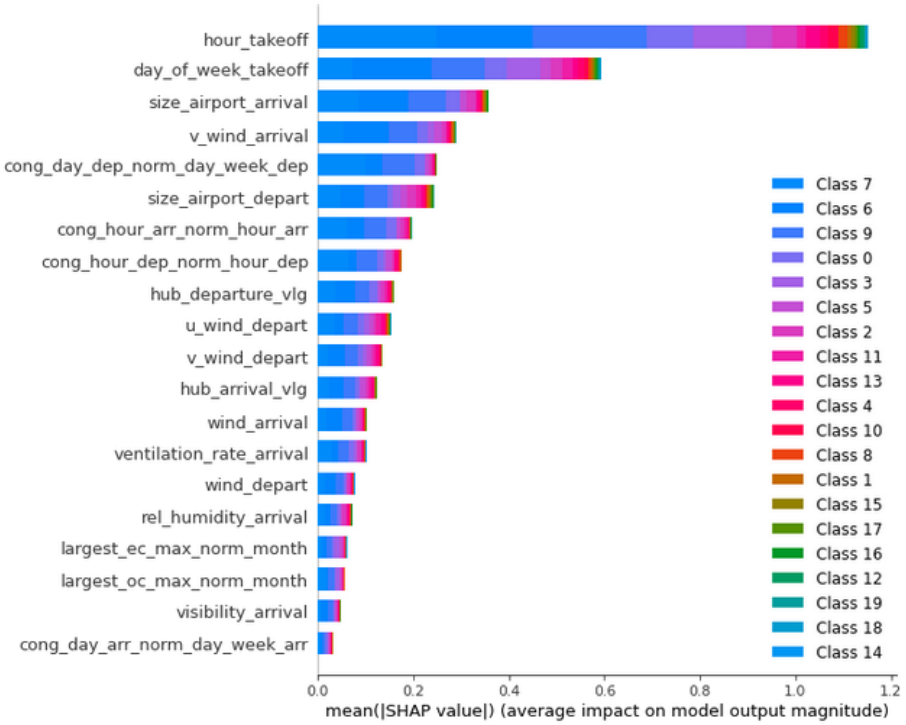



**Figure VI-21:** *SHAP analysis multi-output classifier ATFM delay*

## VI.6 Advice capabilities

Two advice generators are proposed for predicting ATFM regulations at the flight level. First, it is proposed a simple integrated view of the results from the different models for specific flights. This could help to easily see the outcome of the different models and the uncertainty of the predictions. Second, it is proposed to visualize the estimated ATFM delay along the different rotations plan for a specific day and aircraft frame. This will help to see the severity and impact of the estimated ATFM delay and the possible reactionary delay.

### VI.6.1 Integration view

An integrated view that provides information about the expected ATFM regulations that can be issued for specific OD pairs is proposed to support the operational plan definition phase and overcome possible downstream effects of ATFM regulations. ATFM delay can severely impact the airlines' fleet performance as they are typically issued around 4 hours before departure, but they can fluctuate until CTOT. As mentioned at the beginning of this Chapter, the prediction horizon of the models proposed is around 24 hours before Estimated Off-Block Time (EOBT), anticipating possible ATFM regulations of future flights.

Figure VI-22 shows the outcome of the advice generator for a flight that is expected to be regulated with a ATFM different than zero. In this example, the models show high confidence in the predictions showing all the results in green, issuing an expected delay of five minutes.

However, when the models present a lower confidence level in the predictions, to clearly show to the end user (e.g., the duty manager) that the models are less sure about the expected outcome, the labels are displayed in red. Figure VI-23 exhibits an example where the models are not very sure about whether the flight is going to be regulated or whether the ATFM delay will be zero. Despite the low confidence level, the information is still useful for the duty manager, indicating that the flight should be monitored.

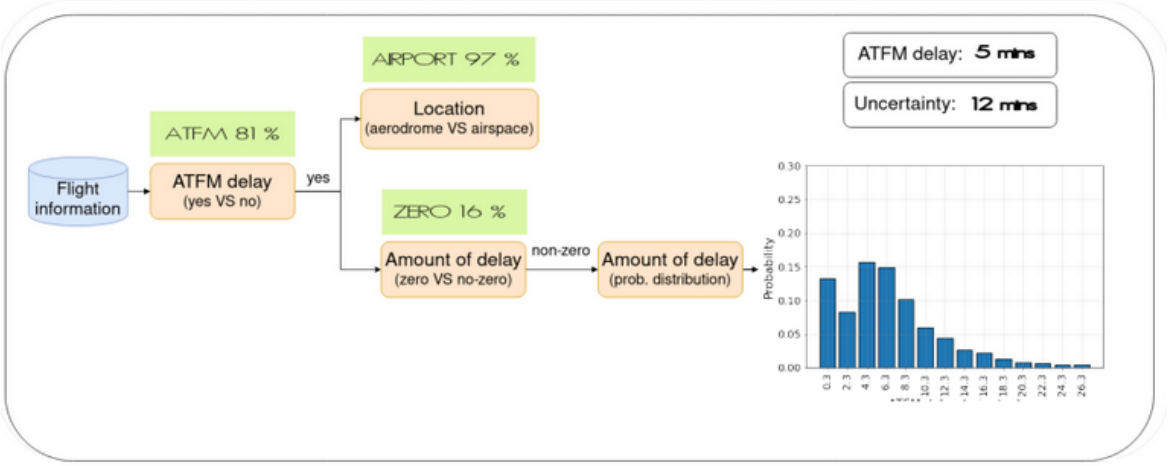



**Figure VI-22:** *Example certain prediction regulated flight with non-zero ATFM delay*

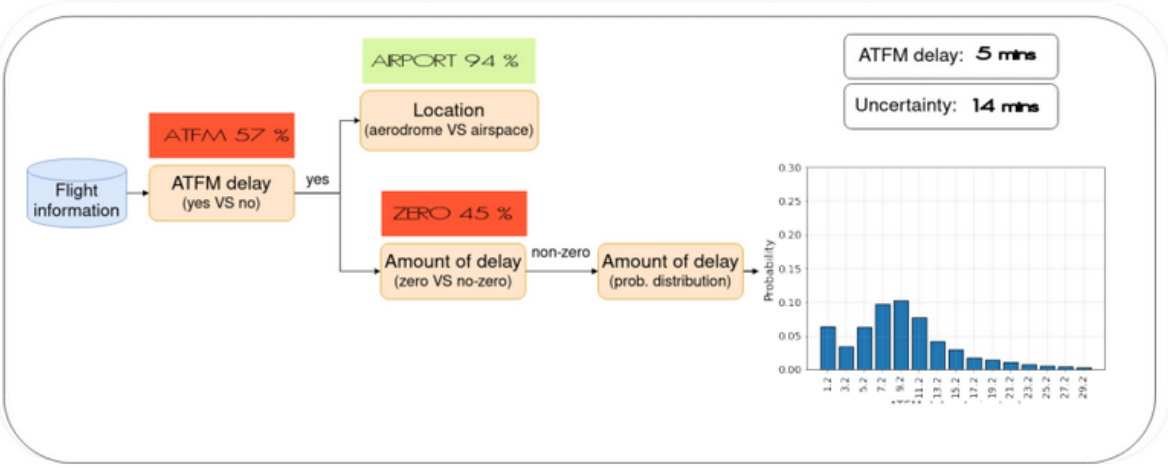



**Figure VI-23:** *Example uncertain prediction regulated flight with non-zero ATFM delay*

Finally, as previously mentioned, it is crucial to show the right level of information at each moment to provide meaningful advice. To do so, in those cases where the models are predicting the negative classes with high confidence, the integration view is limited to show this characteristic. Figure VI-24 shows the integrated view for a flight that is expected to have zero minutes of ATFM delay. Notice that the probability distribution is not displayed in this example, as the expected delay is zero.

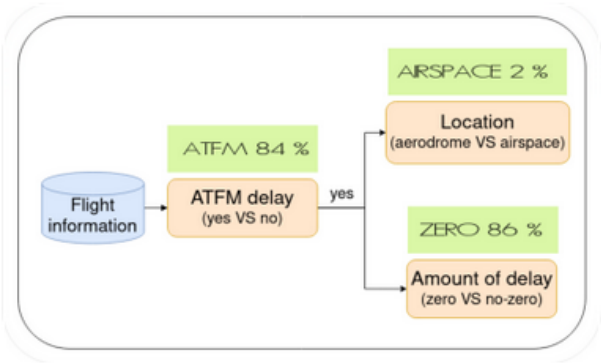



**Figure VI-24:** *Example certain prediction regulated flight with zero ATFM delay*

### VI.6.2 Reactionary delay

The second advice generator proposed for the visualization of ATFM regulations at the flight level is based on integrating the ATFM models into a system able to provide visual and numerical advice for the planned rotation of a particular aircraft frame.

A convolutional process based on a do-nothing approach propagates ATFM delay along the different rotations of an aircraft for a particular day of operations. The do-nothing means that delay will be propagated without measures to prevent their excessive propagation from being modeled. This means that the system will be able to highlight when undesirable situations (e.g., breaching a curfew or missing an ATFM slot) might occur, prompting the duty manager to act. Note that for this approach to estimate the probability of these undesirable outcomes successfully, the expected probability distribution of delays must be predicted. The reader is referred to Section II.7.4 for further details.

The reactionary delay model will get the fleet status at a given moment in time, gathering information on flights flown, being operated, and planned for a given aircraft frame. This means that at a given moment, the flight being considered might already be delayed (primary or accrued delay up to that moment), and successive rotations might already be regulated (or not). If flights are already regulated (or too close in time < 4 hours from current time), the information on their ATFM status is considered fixed as in the fleet status obtained. For the remaining flights, the machine learning models of ATFM delay are used to estimate their probability of being regulated and the amount of delay experienced. With this information and estimating block time and minimum turnaround times (to estimate the earliest possible aircraft ready time), delays are propagated as a convolution of the departure, operating, and arriving stochastic processes.

As an example, Figure VI-25 displays the expected rotations for the registration mark ECMCU, assuming the system was triggered on 12/09/2018 at 7h00. As can be seen, the ATFM delay assigned to the third rotation (EDDH-LEBL) produces a downstream effect introducing some reactionary delay in the fourth rotation (LEBL-LEZL), increasing the probability of missing the ATFM slot up to 25%. This information could be used to consider, for instance, an aircraft swap to reduce the probability of missing the slot. Another option could be to ask the NM for a later slot and cancel the last rotation. In any case, the final action depends on the needs or policies of the airline.

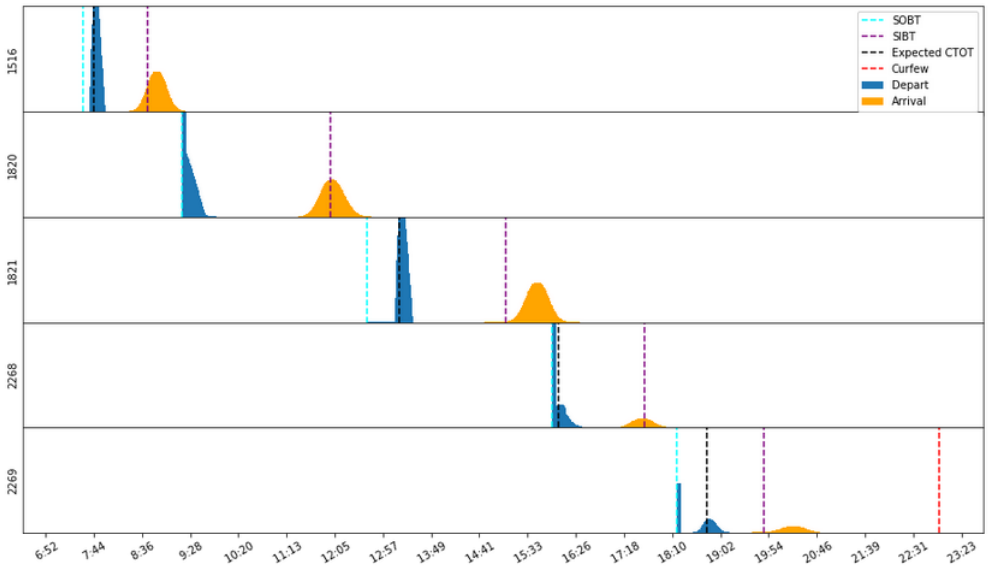



**Figure VI-25:** *Visual outcome of the reactionary delay system for the registration mark ECMCU*

Something very interesting about this approach is that it is possible to see the outcome of the intermediate processes thanks to the usage of probability distributions rather than the expected values. Therefore, they can be used to complement the previous advice. Figure VI-26 shows the arrival time distribution for the rotation EDDH-LEBL and the SIBT (purple line). Then, Figure VI-27 shows the expected minimum turnaround time at LEBL before the departure to LEZL. Finally, Figure VI-28 exhibits the expected aircraft ready time, the expected CTOT (black line), and the time until missing the slot (red). Notice that all the values of the aircraft ready at the right of the time to miss the ATFM slot are the probability for the flight to miss this ATFM slot. Missing this slot will require requesting a new one which could induce significant additional delay and, potentially, losing the slot on the returning leg (LEZL-LEBL flight).

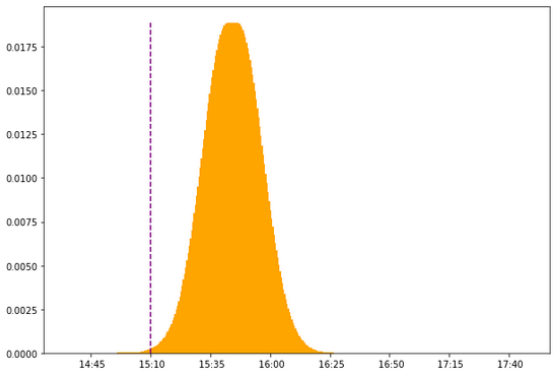



**Figure VI-26:** *Arrival time distribution EDDH-LEBL flight*

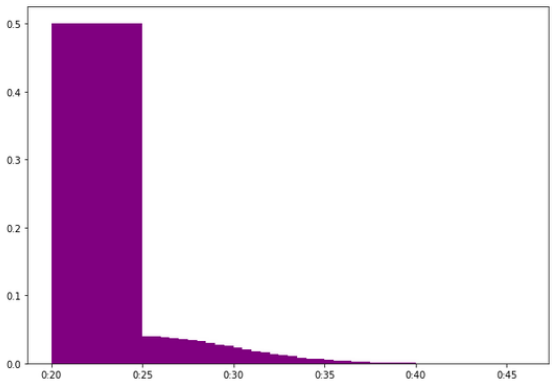



**Figure VI-27:** *Minimum turnaround time at LEBL*

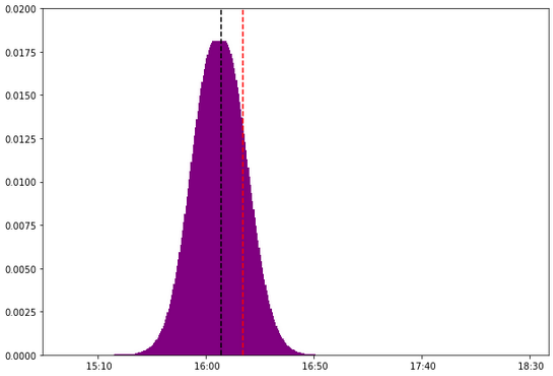



**Figure VI-28:** *Aircraft ready time for LEBL-LEZL flight*

Similar results can be obtained for any aircraft frame from which we have the planned rotations. Moreover, this framework could be used to identify other non-observable actions in the historical datasets, such as aircraft swaps, cancellations, or breaching a curfew.

## VI.7 Discussion

This Chapter studied the usage of ML models to predict ATFM characteristics at the flight level(*i.e.*, for individual flights). Concretely, if it is feasible to predict the *probability* of ATFM regulation, the *protected location*, whether the ATFM delay is going to be *zero*, and the *ATFM delay*. For the airlines, the previous information is crucial when deciding if further actions are required to avoid possible downstream effects.

The results show that it is possible to accurately estimate the probability and protected location of ATFM 24 hours before departure when actual ATFM information is not available. However, predicting the exact minutes of ATFM delay is a much more challenging problem using only information available by the airlines. Predicting whether the ATFM delay is going to be zero reported accuracy of 0.69, and the ATFM delay a MAE of 9.5 minutes. The prediction of the exact minutes of delay is very complex as it depends on the CASA, which is a simple (but fair) system based on the principle of first-in-first-serve. Therefore, it does not depend on the OD pair nor the congestion of the network, as was presented in the future correlation analysis (see Section VI.4.2). It mainly depends on when the flight is expected to cross the congested region. Furthermore, the model has to distinguish zero and very small amounts of ATFM delay.

Despite some of the limitations found, the overall performance of the framework is between 70% and 88%, clearly indicating that it is possible to predict ATFM information for individual flights. One important lesson learned in this Chapter is the need to consider the predictions' uncertainty. Concretely, it has been used the combination of a regressor and a classifier to predict the probability distribution of ATFM delay increases the advice capabilities of the system. However, it would be interesting to compare the results obtained when predicting the probability distribution with more conventional approaches, such as NGBoost or CatBoost with RMSEwithUncertainty.

The integration of the models could allow the identification of non-observable actions in the historical datasets, such as missing the ATFM slot or breaching a curfew. Further study is required when integrating the ML models to provide advice on reactionary delays, but initial promising results have been obtained.

Finally, two operational constraints are identified in this Chapter. First, using data sources available for the end user is paramount, making the prediction of the exact ATFM delay difficult. Second, the fact that the models are developed for a specific operator limits their possible deployment. In theory, the models should only be used to identify ATFM characteristics for flights flown by the selected airline. However, Europe is a very regulated environment, and therefore, the behavior of different airlines should be very similar, indicating that the models could be extended to other airlines.

Lastly, Appendix B summarizes and compares the results obtained using the best possible pre-tactical information available (Last filled pre-tactical flight plans and actual weather information). Notice that this information has not been used in this Chapter as their availability is not guaranteed in the defined prediction horizon. However, the results show that the best results are not always obtained using perfect data. The best option is to use the same data available when the actions were recorded.

# VII

# Concluding Remarks

The Air Traffic Management (ATM) system is reaching its capacity limit, and the expected continued traffic growth indicates that the delay situation will deteriorate drastically if bold actions are not taken. In Europe, the Single European Sky Air traffic management Research (SESAR) program addresses the impact of air traffic growth by studying novel procedures and technologies, aiming to improve information sharing and the levels of automation. In this thesis, Machine Learning (ML) techniques and their use for pre-tactical delay advice were studied to support the different stakeholders, taking into account the native uncertainty in the models.

During this work, some questions arose that were assessed; some are still open and could be further research topics. A summary and conclusions of the achieved results and hints on the possible directions for future work are presented in what follows.

## VII.1 Summary of Contributions

The main contributions of this PhD thesis are summarized as follows:

- An software architecture for robust and consistent experimentation was presented in Chapter II. The framework is divided into three well-known layers: data infrastructure, predictive capabilities, and advice capabilities. It aims to use a flexible data lake to study the best possible data sources to predict/solve Air Traffic Flow Management (ATFM) regulation during the pre-tactical phase. Moreover, specialized software architectures to provide advice for different stakeholders were presented;

- The previous architecture was used in Chapter III to study whether it was feasible to predict airspace ATFM regulation at the Traffic Volume (TV) level, to create a support tool for the Network Manager (NM). Different input features, scalar variables, images, and different ML algorithms were used. Moreover, it studied the usage of different evaluation techniques according to the desired granularity in the results. The most important conclusion of this study was that it is possible to predict C-ATC Capacity regulation with accuracy higher than 80% in the most regulated TVs from the most regulated European regions. Furthermore, it is shown the importance of proper advice capabilities when showing the results of ML models. In this case, the advice generator focused on representation fidelity with respect to the current tools used. Although the tools aim to be used by the NM, the airlines also could take advantage of pre-tactically knowing congested en-route sectors;

- To extend the previous work, Chapter IV investigated whether the proposed architecture can be used to predict other ATFM regulations reasons. In this case, en-route W-Weather regulations due to convective weather. Promising results were obtained, and more than 80% of the regulations were precisely identified. Moreover, similar to the previous experiment, the behavior of the models was validated using eXplainable Artificial Intelligence (XAI) techniques;

- After the detection phase, it is required to smooth the expected traffic to ensure that demand meets the predefined capacity of the TVs. To do so, Chapter V studied the use of Reinforcement Learning (RL) techniques to automatize the resolution of identified ATFM regulations. Two different types of algorithms were studied with different configurations. Although the results did not clearly conclude which algorithm configuration best fits the problem, algorithms based on continuous actions arise as promising candidates. The agents have more freedom when cooperatively deciding on required ground delays. Furthermore, it has been proved that it is possible to use images to overcome scalability issues identified in the literature;

- Airlines are the airspace users mainly affected by ATFM regulations. Continuous monitoring and submission of new flight plans are key to optimizing the different rotations of flights during day of operation (D0). Chapter VI studied whether it is also possible to predict ATFM regulations and their characteristics at the flight level. To this end, four different supervised ML models were developed. The results indicate that predicting the probability of regulations and the protected location is feasible using ML models. However, predicting the expected ATFM delay is much more challenging. This is because ground delay is imposed by the Computer Assisted Slot Allocation (CASA) based on the principle of first-in-first-serve. To predict such information, the airline should have information about the intention of all other operators and the moment the flight will enter the congested region. The main conclusions of this work are that it is possible to accurately predict ATFM at the flight level, and more importantly, the models can be used to estimate possible reactionary delays along for specific aircraft frames, which could be key to identify situation such as missing departing slots or breaching curfew.

## VII.2 Future Research

During this PhD thesis, new questions and research lines arose. Taking advantage of the ML models, architecture, and approaches that have been developed, several work items that deserve further research and/or resources have been identified. The following elements could potentially improve the solutions proposed:

- The combination of the work presented in Chapter III, Chapter IV, and Chapter V would create a fully automated system able to identify required en-route ATFM regulations and provide advice on their resolution. The behavior, advantages, and limitation of a fully automated system could be really interesting for the aviation community. Especially from the perspective of the NM and the airlines, taking advantage of the possible early detection of ATFM regulations;

- However, the identification of ATFM regulations is limited to two specific regulations reasons. Training new models to identify other regulation reasons could enrich the outcome of the proposed framework. However, additional data sources will be required, e.g., sources related to the staff available or possible military actions;

- Another possible future research line could be the study of specialized VS global models. As presented in the experiments at the TV level, the best performance was obtained using specialized models. However, this approach could create scalability issues when deploying models for the entire European Civil Aviation Conference (ECAC) region. Nonetheless, it has been proved that it is possible to train models to identify regulation over specific regions rather than unique TVs. Combining these two approaches could be the key to creating a system that could be used for industrialization. Ideally, it would be incredibly useful to develop a system able to apply ATFM regulations over the entire ECAC area;

- One of the major challenges in this thesis was data availability. Therefore, another possible research could be centered on studying the models' performance as a function of dataset size, trying to see whether the proposed architecture provides better or worse results. Moreover, it will be really interesting to study the selected approach avoiding down-sampling the datasets, testing the models in more realistic conditions, and having to deal with an unbalanced number of non-regulated and regulated observations;

- On the other hand, during the resolution phase using RL techniques, the developed reward function was centered on minimizing the overall delay. However, it could be really interesting to take into account other indicators, such as the expected weather conditions or costs. Weather information could be integrated into the images, while the cost of the actions could be encoded in the reward function directly;

- Also related to the resolution approach, the proposed tool is centered on a specific TV without taking into account the congestion of adjacent sectors. In future work, the impact/effect of neighboring sectors/TVs could be tested. All this, introducing departing windows rather than using exactly the Scheduled Off-Block Time (SOBT) as the take-off time;

- For the prediction of ATFM characteristics at the flight level, future research could be the study and development of new input features to improve the accuracy of the models that estimate the issued delay. To do so, the moment the flight enters the congested sectors must be taken into account. A good starting point could be to combine the work from Chapter III and Chapter VI to identify whether a flight is going to be regulated, then try to predict which sector is congested, and finally combine this information with the expected flight plan to predict the possible ground delay. Additionally, it would be interesting to try a temporal split during the training and testing of the models, comparing the results with the ones reported in this thesis;

- Finally, one specific operator has been selected to predict ATFM characteristics on the planned rotations. However, studying whether the models can be extrapolated to other airlines or the impact of having a system as this deployed could be critical to determine if it is a valid approach for industrialization.

# A

# ATFM regulations at TV level - Spanish case

In Chapter III and Chapter IV, it has been proved that it is possible to predict C-ATC Capacity and W-Weather Air Traffic Flow Management (ATFM) regulations using the proposed *RNN-CNN* cascade time-distributed architecture which combined a time-distributed Recurrent Neural Network (RNN) and a Convolutional Neural Network (CNN).

In this Appendix, results from a different region are provided to validate the ability of the system to be deployed in other regions. Traffic Volumes (TVs) from Spain have been selected because it is a region with a considerable number of regulations but less challenging than MUAC and REIMS.

Results using the same approach, methodology, and techniques are obtained to study the performance of the mentioned models in this new region. Thus, only the final results are provided for completeness. Section A.1 shows the results obtained predicting C-ATC Capacity regulations, while Section A.2 focuses on identifying W-Weather regulations.

## A.1 C-ATC Capacity ATFM regulations

Table A-1 shows the accuracy, recall, precision, and F1-Score predicting C-ATC Capacity regulation in the top-three most regulated TV in Spain. As can be seen, the models are able to accurately predict the time-steps that should be regulated with an accuracy of around 85% and 80% for the specialized and global models, respectively. Compared to the results obtained in MUAC and REIMS, the results show a drop between 2% and 4%. On the other hand, the interval analysis

reveals that the models can identify all the regulations too.

**Table A-1:** *Performance RNN-CNN cascade model for en-route C-ATC Capacity regulations*

| | | | Time-Step Classification | | | | Interval Classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region** | **TV** | **Train/Test** | **Accuracy** | **Recall** | **Precision** | **F1-Score** | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| Spain | BAS | 351/136 | 89.47 | 95.75 | 84.01 | 88,49 | 90.58 | 100 | 85.82 | 92.36 |
| | CCC | 234/102 | 85.35 | 91.37 | 79.14 | 84.81 | 85.82 | 100 | 79.73 | 88.72 |
| | BLI | 267/113 | 82.78 | 87.34 | 82.62 | 84.91 | 84.23 | 100 | 84.68 | 91.81 |
| | All | 998/373 | 78.54 | 84.46 | 81.19 | 82.92 | 85.99 | 100 | 78.38 | 87.93 |

After the training, the confidence level and the SHapley Additive exPlanations (SHAP) analysis exhibit very similar behavior in all the studied TVs across all the regions (Spain, MUAC, and REIMS), indicating that the proposed architecture can be extrapolated to other regions with no difficulty. The only operational constraint is the reduced number of samples to train the models.

## A.2 W-Weather ATFM regulaitons

Table A-2 presents the accuracy, recall, precision, and F1-score obtained to predict W-Weather ATFM regulation in Spanish TVs. The overall performance of the models is around 80% for the time-step analysis, indicating that the proposed architecture could also be used to precisely identify W-Weather regulations in other regions without difficulties. The most interesting result of this study is that the interval analysis no longer reports a 100% recall in all the scenarios. For one of the specialized and the global models, it can be observed a 2% drop indicating that the framework has not been able to identify all the regulated intervals. However, as mentioned previously, the overall performance is still promising.

**Table A-2:** *Performance RNN-CNN cascade model for en-route W-Weather regulations*

| | | | Time-Step Classification | | | | Interval Classification | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Region** | **TV** | **Train/Test** | **Accuracy** | **Recall** | **Precision** | **F1-Score** | **Accuracy** | **Recall** | **Precision** | **F1-Score** |
| Spain | CCC | 325/117 | 85.87 | 89.45 | 80.21 | 84.57 | 86.71 | 100 | 80.83 | 89.39 |
| | LVU | 213/95 | 81.54 | 88.62 | 77.87 | 82.88 | 82.92 | 100 | 77.81 | 87.18 |
| | DI1 | 242/103 | 77.59 | 84.56 | 79.58 | 81.99 | 80.48 | 99.21 | 80.12 | 93.25 |
| | All | 976/351 | 74.87 | 78.91 | 79.74 | 19.3 | 81.18 | 98.37 | 74.35 | 84.55 |

The model explainability analysis shows that this new set of models has very similar behavior to the previous ones for the MUAC and REIMS regions. Only minor variations can be observed.

## A.3 Discussion

This experiment aimed to transfer the proposed architecture and approach to other TVs in the European Civil Aviation Conference (ECAC) region. Promising results have been observed with consistent accuracy, recall, precision, and F1-score between TVs independently of the region. This clearly shows that, if desired, the methodology proposed has a good level of scalability. Therefore, the research question presented in this Appendix has been positively answered, and the novel architecture to predict C-ATC and W-Weather ATFM regulations at the TV level could be considered for industrialization.

# B

# ATFM regulations at the flight level - Perfect data

Many data sources are available containing Air Traffic Management (ATM) information. However, as previously mentioned, it is almost impossible to know what information is available at each moment. Therefore, it is almost impossible to know precisely the available information in the selected prediction horizon: day prior to operations (D-1).

First, results were obtained using what was called "perfect data". That is, the *Last filled pre-tactical flight plan* and *actual weather information*. This experiment aimed to identify which of the proposed Air Traffic Flow Management (ATFM) characteristics at the flight level is feasible to predict from a Machine Learning (ML) perspective.

This experiment presents some operational constraints, mainly related to data availability issues. Actual weather information is not available on D-1 nor all the flight plans. On D-1 it is available the information used in Chapter VI: Flight Intention (FI) and weather forecast. However, some really interesting results were found that could add value to this thesis.

## B.1 Data sources

Table B-1 collects the data sources used for this experiment where "perfect data" is assumed to be available at the prediction horizon D-1. As a source of network data, ALLFT+ data is used to know the flight intention and the flight plans. Concretely, it has been used *M1 traffic*, which corresponds with the last filed flight plan filled by the airlines before any regulation was applied. Then, the actual convective weather recorded at the airports is used as a source of weather information.

Section B.1.1 provides more details about this new weather data source.

**Table B-1:** *Data sources used to predict ATFM regulations for individual flights (flight level)*

| Data source | Period time | Usage | Comment |
|---|---|---|---|
| ALLFT+ | 2018 | Features | Flight intentions and Features |
| Airports data | Static | Features | Size and/or hub |
| METAR | 2018 | Features | Weather |
| Vueling | 2018 | Labelling | ATFM information |

### B.1.1 METAR

Typically, METeorological Aerodrome Report (METAR) information comes from airports or permanent weather observation stations and is reported every half-hour. This information is encoded and standardized to provide information as precisely as possible. Raw METAR data is the most common format for transmitting observational weather data. Table B-2 describes the most relevant features used from this data source.

**Table B-2:** *METAR most relevant weather-related features*

| Name | Description | Units |
|---|---|---|
| Airport | International Civil Aviation Organization (ICAO) airpot code | Dimensionless |
| U-component wind | Eastward component of the wind | $ms^{-1}$ |
| V-component wind | Northward component of the wind | $ms^{-1}$ |
| Wind | Nominal wind speed | $s^{-1}$ |
| Visibility | Prevailing visibility[1] | $m$ |
| RVR | Runway Visibility Range | $m$ |
| Snow | Snow is falling at a heavy intensity | Dimensionless |
| Temperature | Temperature in the atmosphere | $F$ |
| Runway | Condition of the runway | Dimensionless |
| CAVOK | Ceiling And Visibility OK (no cloud below 5,000 ft) | Boolean |

It is worth mentioning that a specific decoder is required to extract the meteorological information from METAR data. This decoder has not been implemented in this thesis.

## B.2 Probability ATFM delay

Table B-3 contains the results using "perfect data", the ones reported in Chapter VI, and the difference in performance between the two approaches when predicting the probability of ATFM regulations. The same methodology, approach, and labeling has been used in both cases. As can be seen, there is a drop between 5% and 10% in the performance using the forecasts.

**Table B-3:** *Accuracy, recall, precision, F1-Score probability ATFM delay*

| Approach | Accuracy | Recall | Precision | F1-score |
|---|---|---|---|---|
| Perfect data | 0.89 | 0.91 | 0.86 | 0.89 |
| Forecast | 0.82 (-0.07) | 0.81 (-0.1) | 0.82 (-0.04) | 0.82 (-0.07) |

The drop in the performance could come from the fact that the features correlation analysis reported a higher correlation of the network demand features than using the forecast. However, using the forecast, the most correlated features are related to possible convective weather, which is a less frequent regulation reason.

## B.3 Location ATFM regulation

This Section summarizes the result obtained predicting the protected location region of ATFM regulations. Table B-4 contains the results using "perfect data", the ones reported in Chapter VI, and the difference in performance between the two approaches using the same methodology, approach, and labeling has been used. It is very interesting to see that for this case study, the performance of the models based on using forecast is better than using "perfect data".

**Table B-4:** *Accuracy, recall, precision, F1-Score location ATFM regulations*

| Approach | Accuracy | Recall | Precision | F1-score |
|---|---|---|---|---|
| Perfect data | 0.83 | 0.81 | 0.87 | 0.83 |
| Forecast | 0.87 (+0.04) | 0.84 (+0.06) | 0.89 (+0.02) | 0.86 (+0.03) |

The improvement in the performance seems to come from a higher correlation of the input features to the target labels. Both analyses reported similar results, but using forecast, the scores obtained were around 5% larger than using "perfect data". This case study is a clear example of the importance of using data sources equivalent to the ones available when the actions were recorded. Ultimately, the goal of using supervised machine learning models is to replicate past decisions in future scenarios.

## B.4 Zero VS Non-Zero delay

Table B-5 shows the results obtained between using "perfect data" and forecasts, using the same methodology, approach, and labeling in both cases. Similar to the probability of ATFM delay, there is a drop between 5% and 10% in the performance of the model that uses the forecast. However, in this case, study, the drop in performance is more critical as the overall accuracy of the models is lower.

The correlation and SHapley Additive exPlanations (SHAP) analysis reported similar results in both case studies. The engineered input features and target labels have low correlation and activation values.

**Table B-5:** *Accuracy, recall, precision, F1-Score zero minutes ATFM delay*

| Approach | Accuracy | Recall | Precision | F1-score |
|---|---|---|---|---|
| Perfect data | 0.75 | 0.72 | 0.8 | 0.76 |
| Forecast | 0.69 (-0.06) | 0.67 (-0.05) | 0.69 (-0.11) | 0.69 (-0.07) |

## B.5 ATFM delay distribution

Finally, Table B-6 compares the results of both models using "perfect data" and the forecasts, estimating the minutes of ATFM delay. The results in this last case study do not follow a clear pattern. As can be seen, the is a drop in the precision of the model identifying the exact ATFM delay but an improvement in the uncertainty the model reports with equivalent number of hits (actual ATFM delay within the predicted probability distribution).

**Table B-6:** *MAE, uncertainty, and number of hits ATFM delay*

| Approach | MAE (mins.) | Mean 90% probability (mins.) | Hits (%) |
|---|---|---|---|
| Perfect data | 9.35 | 14.60 | 0.87 |
| Forecast | 9.58 (+0.23) | 12.87 (-1.73) | 0.88 (+0.01) |

The feature correlation and SHAP analysis do not show a clear pattern of behavior. The most interesting result is that the models predict that most of the ATFM will be around five minutes. Therefore, a possible future work could be to change the zero VS non-zero delay models for a less VS more than five minutes of delay.

## B.6 Discussion

This experiment aims to see the impact of the data sources when predicting ATFM characteristics at the flight level. With this objective, two sets of data sources have been used: "perfect data" and forecast for D-1. The results do not show clear evidence of what approach is better. New trends in ML suggest training the models with the best possible data and then using the best available data in deployment, even though they are different. However, the author rejects this approach because there is a loss of control over the expected behavior of the models, which is unacceptable in safety-critical environments.

# Bibliography


ABDI, HERVÉ, & WILLIAMS, LYNNE J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, **2**(4), 433–459. 5

ABU-MOSTAFA, YASER S, MAGDON-ISMAIL, MALIK, & LIN, HSUAN-TIEN. 2012. *Learning from data*. Vol. 4. AMLBook New York, NY, USA. 51

AGOGINO, ADRIAN, & TUMER, KAGAN. 2009. Learning indirect actions in complex domains: action suggestions for air traffic control. *Advances in complex systems*, **12**(04n05), 493–512. 88

AIR TRANSPORTATION ACTION GROUP, ATAG. 2019 (May). *Key facts and figures from the world of air transport*. Tech Report. Air Transportation Action Group, Geneva, Switzerland. 1

ARULKUMARAN, KAI, DEISENROTH, MARC PETER, BRUNDAGE, MILES, & BHARATH, ANIL ANTHONY. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, **34**(6), 26–38. 21

BARNHART, CYNTHIA, BERTSIMAS, DIMITRIS, CARAMANIS, CONSTANTINE, & FEARING, DOUGLAS. 2012. Equitable and efficient coordination in traffic flow management. *Transportation science*, **46**(2), 262–280. 50

BASODI, SUNITHA, JI, CHUNYAN, ZHANG, HAIPING, & PAN, YI. 2020. Gradient amplification: An efficient way to train deep neural networks. *Big data mining and analytics*, **3**(3), 196–207. 54

BASORA, LUIS, MORIO, JÉRÔME, & MAILHOT, CORENTIN. 2017. A trajectory clustering framework to analyse air traffic flows. *In: In 7th SESAR Innovation Days (SIDS)*. Belgrade , Serbia: SESAR Joint Undertaking. 13, 55, 85

BEN-GAL, IRAD. 2008. Bayesian networks. *Encyclopedia of statistics in quality and reliability*. 19

BERTSIMAS, DIMITRIS, & PATTERSON, SARAH STOCK. 1998. The air traffic flow management problem with enroute capacities. *Operations research*, **46**(3), 406–422. 4

BISHOP, CHRISTOPHER M. 2006. *Pattern recognition and machine learning*. Springer. 19, 50

BOLIĆ, TATJANA, CASTELLI, LORENZO, COROLLI, LUCA, & RIGONAT, DESIRÉE. 2017. Reducing ATFM delays through strategic flight planning. *Transportation Research Part E: Logistics and Transportation Review*, **98**, 42–59. 84

BUIZZA, ROBERTO, HOUTEKAMER, PL, PELLERIN, GERALD, TOTH, ZOLTAN, ZHU, YUEJIAN, & WEI, MOZHENG. 2005. A comparison of the ecmwf, msc, and ncep global ensemble prediction systems. *Monthly weather review*, **133**(5), 1076–1097. 32

BURTON-JONES, ANDREW, & GRANGE, CAMILLE. 2013. From use to effective use: A representation theory perspective. *Information systems research*, **24**(3), 632–658. 68

CHAN, MUN CHOON, & LIN, YOW-JIAN. 2005 (May). Behaviors and effectiveness of rerouting: A study. *In: IEEE International Conference on Communications (ICC)*. IEEE, Seoul, South Corea. 4

CHATTERJI, GANO, & SRIDHAR, BANAVAR. 2001. Measures for air traffic controller workload prediction. *In: 1st AIAA, Aircraft, Technology Integration, and Operations Forum*. Los Angeles, CA, USA: AIAA. 5

CHEN, YUTONG, XU, YAN, HU, MINGHUA, & YANG, LEI. 2021. Demand and Capacity Balancing Technology Based on Multi-agent Reinforcement Learning. *In: 40th Digital Avionics Systems Conference (DASC)*. Portsmouth, VA, USA: AIAA/IEEE. 50

CHENG, CHENG, GUO, LIANG, WU, TONG, SUN, JINLONG, GUI, GUAN, ADEBISI, BAMIDELE, GACANIN, HARIS, & SARI, HIKMET. 2021. Machine-Learning-Aided Trajectory Prediction and Conflict Detection for Internet of Aerial Vehicles. *IEEE Internet of Things Journal*, **9**(8), 5882–5894. 50

COOK, ANDREW J, & TANNER, GRAHAM. 2015 (Dec). *European airline delay cost reference values*. Tech Report. University of Westminster, London, UK. 4, 10, 103

COOK, ANDREW J, TANNER, GRAHAM, & ANDERSON, STEPHEN. 2004 (May). *Evaluating the true cost to airlines of one minute of airborne or ground delay*. Tech Report. EUROCONTROL, Performance Review Commission, Brussels, Belgium. 7

CORRADO, SAMANTHA J, PURANIK, TEJAS G, PINON, OLIVA J, & MAVRIS, DIMITRI N. 2020. Trajectory clustering within the terminal airspace utilizing a weighted distance function. *Proceedings*, **59**(1), 7. 85

CORTES, CORINNA, & VAPNIK, VLADIMIR. 1995. Support-vector networks. *Machine learning*, **20**(3), 273–297. 19

COTON. 2018. *Capacity Optimisation in Trajectory-Based Operations*. https://www.sesarju.eu/projects/cotton. Accessed: September 7, 2022. 6

CRESPO, ANTONIO MARCIO FERREIRA, WEIGANG, LI, DE BARROS, ALEXANDRE GOMES, *et al.* 2012. Reinforcement learning agents to tactical air traffic flow management. *International journal of aviation management*, **1**(3), 145–161. 84

CRUCIOL, LEONARDO LBV, DE ARRUDA JR, ANTONIO C, WEIGANG, LI, LI, LEIHONG, & CRESPO, ANTONIO MF. 2013. Reward functions for learning to control in air traffic flow management. *Transportation research part c: Emerging technologies*, **35**, 141–155. 89

DALMAU, RAMON. 2022. Predicting the likelihood of airspace user rerouting to mitigate air traffic flow management delay. *Transportation research part c: Emerging technologies*, **144**, 103869. 38, 84, 98

DALMAU, RAMON, ZERROUKI, LEILA, ANOURAUD, CAMILLE, SMITH, DARREN, & CRAMET, BENJAMIN. 2021a. Are all the requested air traffic flow management regulations actually indispensable? *In: 11th SESAR Innovation Days (SIDS)*. Virtual Event: SESAR Joint Undertaking. 10

DALMAU, RAMON, GENESTIER, BRICE, ANORAUD, CAMILLE, CHOROBA, PETER, & SMITH, DARREN. 2021b. A machine learning approach to predict the evolution of air traffic flow management delay. *In: 14th USA/Europe Air Traffic Management Research and Development*. New Orleans, LA, USA: SESAR Joint Undertaking. 8

DALMAU CODINA, RAMON, BELKOURA, SEDDIK, NAESSENS, HERBERT, BALLERINI, FRANCK, & WAGNICK, SEBASTIAN. 2019. Improving the predictability of take-off times with Machine Learning: A case study for the Maastricht upper area control centre area of responsibility. *In: 9th SESAR Innovation Days (SIDS)*. Athens , Greece: SESAR Joint Undertaking. 74

DART. 2019. *Data-driven aircraft trajectory prediction research*. https://www.sesarju.eu/index.php/projects/dart. Accessed: September 7, 2022. 6, 50

DATABRICKS. 2022. *Databricks*. https://www.databricks.com/. Accessed: November 9, 2022. 28

DDR. 2022. *Demand Data Repository*. https://www.eurocontrol.int/ddr. Accessed: October 31, 2022. 30

DE FALCO, PAOLINO, & DELGADO, LUIS. 2021. Prediction of reactionary delay and cost using machine learning. *In: Airline group of the International Federation of Operational Research Society (AGIFORS)*. Atlanta, GA, USA: AGIFORS. xi, 10, 35, 37, 45, 109, 110

DE GIOVANNI, LUIGI, LULLI, GUGLIELMO, & LANCIA, CARLO. 2022. Data-driven optimization for Air Traffic Flow Management with trajectory preferences. *arXiv preprint*. 98

DEACON, JOHN. 2009 (Apr). *Model-View-Controller (MVC) architecture*. Tech Report. Computer Systems Development, Consulting & Training, London, UK. 40

DELGADO, LUIS, & PRATS, XAVIER. 2012. En route speed reduction concept for absorbing air traffic flow management delays. *Journal of Aircraft*, **49**(1), 214–224. 84

DELGADO MUÑOZ, LUIS. 2013 (Apr). *Cruise speed reduction for air traffic flow management*. Ph.D. Thesis, Universitat Politècnica de Catalunya (UPC), Castelldefels, Catalonia, Spain. 7

DIETTERICH, THOMAS G. 2002. Machine learning for sequential data: A review. *Pages 15–30 of: Structural, syntactic, and statistical pattern recognition: Joint iapr international workshops sspr 2002 and spr 2002 windsor, ontario, canada, august 6–9, 2002 proceedings*. Springer. 35

DISPATCHER3. 2022 (November). *Innovative processing for flight practices*. Tech Report. University of Westminster, Universitat Politecnica de Catalunya, INNAXIS, PACE aerospace engineering and Information Technology. ix, 23, 44

DISPATCHER3 CONSORTIUM. 2020 (Dec). *D1.1 - Technical Resources and Problem Definition*. Tech Report. Dispatcher3 Consortium. v4.1. 11, 35

DJANGO. 2005. *The web framework for perfectionsims with deadlines*. https://www.djangoproject.com/. Accessed: October 24, 2022. 40

DUAN, TONY, ANAND, AVATI, DING, DAISY YI, THAI, KHANH K, BASU, SANJAY, NG, ANDREW, & SCHULER, ALEJANDRO. 2020. Ngboost: Natural gradient boosting for probabilistic prediction. *Pages 2690–2700 of: International conference on machine learning*. PMLR, Hawaii. 109

ECMRWF. 2022. *European Centre for Medium-Range Weather Forecasts*. https://www.ecmwf.int/. Accessed: October 11, 2022. 32

EDMUNDS, ANGELA, & MORRIS, ANNE. 2000. The problem of information overload in business organisations: A review of the literature. *International Journal of Information Management*, **20**(1), 17–28. 39

EGOROV, MAXIM. 2016 (May). *Multi-agent deep reinforcement learning*. Tech Report. Stanford University Stanford, CA, USA. 86

EUROCONTROL. 2019 (Jul). *ATFM Regulation: a power for good*. Tech Report. EUROCONTROL, Brussels, Belgium. 7

EUROCONTROL. 2020 (Dec). *EUROCONTROL Standard Inputs for Economic Analyses*. Tech Report. EUROCONTROL, Brussels, Belgium. 10

EUROCONTROL. 2022. *ATFM Delay Codes*. https://ansperformance.eu/definition/atfm-delay-codes/. Accessed: September 2, 2022. 8

EUROCONTROL. 2022. *Trajectory-based free routing*. https://www.eurocontrol.int/project/trajectory-based-free-routing. Accessed: August 31, 2022. 10

FAA. 2016 (Aug). *Future NAS*. Tech Report. Federal Aviation Administration, Washington, DC, USA. 9

FAA. 2018 (Apr). *On-time arrival performance*. Tech Report. Federal Aviation Administration, Washington DC, USA. 2

FAA. 2022a (Jul). *Next Generation Air Transportation System (NextGen)*. Tech Report. Federal Aviation Administration, Washington DC, USA. 2

FAA. 2022b. *Trajectory Based Operations (TBO)*. https://www.faa.gov/air_traffic/technology/tbo/. Accessed: August 31, 2022. 10

Fernández, Esther Calvo, Cordero, José Manuel, Vouros, George, Pelekis, Nikos, Kravaris, Theocharis, Georgiou, Harris, Fuchs, Georg, Andrienko, Natalya, Andrienko, Gennady, Casado, Enrique, *et al.* 2017. DART: A machine-learning approach to trajectory prediction and demand-capacity balancing. *In: 7th SESAR Innovation Days (SIDS)*. Belgrade , Serbia: SESAR Joint Undertaking. 84

Flynn, Geraldine, Benkouar, A, & Christien, R. 2003 (Mar). *Pessimistic sector capacity estimation*. Tech Report. EUROCONTROL, Experimental Center, Brétigny-sur-Orge, France. xiii, 3, 5

Foerster, Jakob, Farquhar, Gregory, Afouras, Triantafyllos, Nardelli, Nantas, & Whiteson, Shimon. 2018 (Feb). Counterfactual multi-agent policy gradients. *In: 32nd AAAI conference on artificial intelligence*. 21

Force, EASA AI Task, & Daedalean, AG. 2020 (Mar). *Concepts of Design Assurance for Neural Networks (CoDANN)*. Tech Report. European Union Aviation Safety Agency (EASA), Cologne, Germany. 38

frugally-deep. 2018. *frugally-deep*. https://github.com/Dobiasd/frugally-deep. Accessed: December 19, 2022. 41

Garrigó, Laia, Alsina, Núria, Adrienko, Natalia, Andrienko, Gennady, Piovano, Luca, & Blondiau, Thomas. 2016. Visual analytics and machine learning for air traffic management performance modelling. *In: 6th SESAR Innovation Days (SIDS)*. Delft , Netherlands: SESAR Joint Undertaking. 74

Gianazza, David. 2010. Forecasting workload and airspace configuration with neural networks and tree search methods. *Artificial intelligence*, **174**(7-8), 530–549. 50, 51, 54

Gianazza, David. 2017. Learning air traffic controller workload from past sector operations. *In: 12th USA/Europe Air Traffic Management Research and Development Seminar*. Seattle, United States: ATM Seminar. 5

Gianazza, David, & Alliot, JM. 2002 (Oct). Optimization of air traffic control sector configurations using tree search methods and genetic algorithms. *In: 21st Digital Avionics Systems Conference (DASC)*. AIAA/IEEE, Irvine, CA, USA. 4

Gianazza, David, & Guittet, Kévin. 2006. Selection and evaluation of air traffic complexity metrics. *In: 25TH Digital Avionics Systems Conference (DASC)*. Portland, Oregon, USA: AIAA/IEEE. 50

Gopalakrishnan, Karthik, & Balakrishnan, Hamsa. 2017. A comparative analysis of models for predicting delays in air traffic networks. *In: 12th USA/Europe Air Traffic Management Research and Development Seminar*. Seattle, United States: ATM Seminar. 98

Gorripaty, Sreeta, Hansen, Mark, & Pozdnukhov, Alexey. 2016 (Sep). Decision support framework to assist air traffic management. *In: 35th Digital Avionics Systems Conference (DASC)*. AIAA/IEEE, Sacramento, CA, USA. 98

Graña, Manuel. 2019 (Sep). Dynamic Airspace Configuration: A Short Review of Computational Approaches. *In: International Conference on Computational Collective Intelligence (ICCCI)*. Springer, Hendaye, France. 50

Hafner, Roland, & Riedmiller, Martin. 2011. Reinforcement learning in feedback control. *Machine learning*, **84**(1), 137–169. 22

Hersbach, Hans, Bell, Bill, Berrisford, Paul, Hirahara, Shoji, Horányi, András, Muñoz-Sabater, Joaquín, Nicolas, Julien, Peubey, Carole, Radu, Raluca, Schepers, Dinand, *et al.* 2020. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, **146**(730), 1999–2049. 32

HOCHREITER, SEPP, & SCHMIDHUBER, JÜRGEN. 1997. Long short-term memory. *Neural computation*, **9**(8), 1735–1780. 17

HSU, KIMBERLY. 2014 (Nov). *China's airspace management challenge*. Tech Report. U.S.-China Economic and Security Review Commission, Washington, DC, US. 2

HUANG, CHENG, & XU, YAN. 2021 (Oct). Integrated Frameworks of Unsupervised, Supervised and Reinforcement Learning for Solving Air Traffic Flow Management Problem. *In: 40th digital avionics systems conference (dasc)*. AIAA/IEEE, Portsmouth, VA, USA. 84

IOFFE, SERGEY, & SZEGEDY, CHRISTIAN. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In: International Conference on Machine Learning (ICML)*. Lille, France: PMLR. 90

ISOBAR. 2020. *Artificial intelligence solutions to meteo-based dcb imbalances for network operations planning*. https://www.sesarju.eu/projects/isobar. Accessed: September 7, 2022. 6, 74

ISUFAJ, RALVI, KOCA, THIMJO, & PIERA, MIQUEL ANGEL. 2021. Spatiotemporal graph indicators for air traffic complexity analysis. *Aerospace*, **8**(12), 364. 50

IVANOV, NIKOLA, NETJASOV, FEDJA, JOVANOVIĆ, RADOSAV, STARITA, STEFANO, & STRAUSS, ARNE. 2017. Air traffic flow management slot allocation to minimize propagated delay and improve airport slot adherence. *Transportation research part a: Policy and practice*, **95**, 183–197. 8, 84

JARDINES, ANIEL, SOLER, MANUEL, & GARCÍA-HERAS, JAVIER. 2021. Estimating entry counts and atfm regulations during adverse weather conditions using machine learning. *Journal of air transport management*, **95**, 102109. 74

JUDD, CHARLES M, MCCLELLAND, GARY H, & RYAN, CAREY S. 2017. *Data analysis: A model comparison approach to regression, anova, and beyond*. Routledge. 105

KAMANGIR, HAMID, COLLINS, WAYLON, TISSOT, PHILIPPE, & KING, SCOTT A. 2020. A deep-learning model to predict thunderstorms within 400 km2 South Texas domains. *Meteorological Applications*, **27**(2), e1905. 74

KERAS. 2015. *Keras: simple, flexible and powerful*. https://keras.io/. Accessed: October 24, 2022. 41

KINGMA, DIEDERIK P, & BA, JIMMY. 2015. Adam: A method for stochastic optimization. *In: 3rd International Conference on Learning Representations (ICLR)* . San Diego, CA, USA: ICLR. 90

KOPARDEKAR, PARIMAL, & MAGYARITS, SHERRI. 2003. Measurement and prediction of dynamic density. *In: 5th USA/Europe Air Traffic Management Research and Development Seminar*. Budapest, Hungary: FAA/EUROCONTROL. 5

KRAVARIS, THEOCHARIS, & VOUROS, GEORGE A. 2022. Deep Multiagent Reinforcement Learning Methods Addressing the Scalability Challenge. *In: Multi-agent technologies and machine learning*. London, UK: IntechOpen. 84

KRAVARIS, THEOCHARIS, SPATHARIS, CHRISTOS, BLEKAS, KONSTANTIONS, VOUROS, GEORGE A, & GARCIA, JOSE MANUEL CORDERO. 2018 (Sep). Multiagent reinforcement learning methods for resolving demand-capacity imbalances. *In: 37th digital avionics systems conference (dasc)*. AIAA/IEEE, London, UK. 50

KRIZHEVSKY, ALEX, SUTSKEVER, ILYA, & HINTON, GEOFFREY E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, **60**(6), 84–90. 86

LAMBELHO, MIGUEL, MITICI, MIHAELA, PICKUP, SIMON, & MARSDEN, ALAN. 2020. Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions. *Journal of Air Transport Management*, **82**, 101737. 39

LATTREZ, OLIVIER, MONTES, ROCÍO BARRAGÁN, & MICHALSKI, MATEUSZ. 2022. Predicting Airport ATFM Regulations using Deep Convolutional Neural Networks. *In: 7th SESAR Innovation Days (SIDS)*. Budapest, Hongria: SESAR Joint Undertaking. 74

LILLICRAP, TIMOTHY P, HUNT, JONATHAN J, PRITZEL, ALEXANDER, HEESS, NICOLAS, EREZ, TOM, TASSA, YUVAL, SILVER, DAVID, & WIERSTRA, DAAN. 2015. Continuous control with deep reinforcement learning. *In: 4th International Conference on Learning Representations*. San Juan, Puerto Rico: ICLR. 22, 90

LOWE, RYAN, WU, YI I, TAMAR, AVIV, HARB, JEAN, PIETER ABBEEL, OPENAI, & MORDATCH, IGOR. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, **30**. 21

LUNDBERG, SCOTT M, & LEE, SU-IN. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, **30**. 38

LUO, XIN, WU, HAO, WANG, ZHI, WANG, JIANJUN, & MENG, DEYU. 2021. A novel approach to large-scale dynamically weighted directed network representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**(12), 9756–9773. 39

MAHESH, BATTA. 2020. Machine learning algorithms-a review. *International journal of science and research (ijsr).[internet]*, **9**, 381–386. 19

MARCOS, RODRIGO, ROS, OLIVA G CANTÚ, & HERRANZ, RICARDO. 2017. Combining Visual Analytics and Machine Learning for Route Choice Prediction. *In: 7th SESAR Innovation Days (SIDS)*. Belgrade , Serbia: SESAR Joint Undertaking. 74

MARTÍN MARTÍNEZ, IGNACIO, MATEOS VILLAR, MANUEL, GARCÍA, PEDRO, HERRANZ, RICARDO, GARCÍA CANTÚ-ROS, OLIVIA, & PRATS MENÉNDEZ, XAVIER. 2020. Full-scale pre-tactical route prediction: machine learning to increase pre-tactical demand forecast accuracy. *In: 9th International Conference on Research in Air Transportation (ICRAT)*. Tampa, Florida, USA: ICRAT. 51, 74

MAS-PUJOL, SERGI, SALAMÍ, ESTHER, & PASTOR, ENRIC. 2022. RNN-CNN Hybrid Model to Predict C-ATC CAPACITY Regulations for En-Route Traffic. *Aerospace*, **9**(2), 93. 38

MELGOSA, MARC, PRATS, XAVIER, XU, YAN, & DELGADO, LUIS. 2019. Enhanced demand and capacity balancing based on alternative trajectory options and traffic volume hotspot detection. *In: 38th Digital Avionics Systems Conference (DASC)*. San Diego, CA, USA: AIAA/IEEE. 5, 50

MITCHELL, MELANIE. 1998. *An introduction to genetic algorithms*. MIT press. 19

MITCHELL, TOM M. 1997. *Machine learning*. Vol. 1. McGraw-hill New York. 16

MNIH, VOLODYMYR, KAVUKCUOGLU, KORAY, SILVER, DAVID, GRAVES, ALEX, ANTONOGLOU, IOANNIS, WIERSTRA, DAAN, & RIEDMILLER, MARTIN. 2013. Playing Atari with Deep Reinforcement Learning. *NIPS Deep Learning Workshop* . 89

NIARCHAKOU, S.; SFYROERAS, M. 2022 (Mar). *ATFCM Operations Manua*. Tech Report. EUROCONTROL, Brussels, Belgium. 3, 6, 7, 22, 30, 51, 93

NOAA. 2022. *Global forecast system*. `https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast`. Accessed: November 4, 2022. 33

ODONI, AMEDEO R. 1987. The flow management problem in air traffic control. *In: Flow control of congested networks*. Springer. 50

OXLEY, DAVID, & CHAITAN, JAIN. 2018. *Global Air Passenger Markets: Riding Out Periods of Turbulence*. Tech Report. International Air Transport Association, Montreal, Canada. 1

PARODI, ANTONIO, TEMME, MARCO-MICHAEL, GLUCHSHENKO, OLGA, KERSCHBAUM, MARKUS, SURIAN, NICOLA, BIONDI, RICCARDO, REALINI, EUGENIO, GATTI, ANDREA, TAGLIAFERRO, GIULIO, LLASAT, MARIA CARMEN, *et al.* 2021 (Dec). *H2020 SINOPTICA (Satellite-borne and IN-situ Observations to Predict The Initiation of Convection for ATM) project: initial results*. Tech Report. SESAR Joint Undertaking, Brussels, Belgium. 73

PEETERS, SAM, GUASTALLA, GUGLIELMO, & GRANT, KEVIN. 2018 (Feb). Analysis of en-route vertical flight efficiency. *In: 2018 integrated communications, navigation, surveillance conference (icns)*. IEEE, Brisbane, Australia. 4

PILOT3. 2022 (Jan). *A software engine for multi-criteria decision support in flight management*. Tech Report. University of Westminster, Universitat Politecnica de Catalunya, INNAXIS, PACE aerospace engineering and Information Technology. 43

PLAPPERT, MATTHIAS, HOUTHOOFT, REIN, DHARIWAL, PRAFULLA, SIDOR, SZYMON, CHEN, RICHARD Y, CHEN, XI, ASFOUR, TAMIM, ABBEEL, PIETER, & ANDRYCHOWICZ, MARCIN. 2017. Parameter space noise for exploration. *arxiv preprint*. 91

PRATS, XAVIER. 2011 (Apr). *Contributions to the optimisation of aircraft noise abatement procedures*. Ph.D. Thesis, Universitat Politècnica de Catalunya (UPC), Castelldefels, Catalonia, Spain. ix, 43

PRATS, XAVIER, BARRADO, CRISTINA, NETJASOV, FEDJA, CRNOGORAC, DUSAN, PAVLOVIC, GORAN, AGÜI, IGNACIO, & VIDOSAVLJEVIC, ANDRIJA. 2018. Enhanced indicators to monitor ATM performance in Europe. *In: 8th SESAR Innovation Days (SIDS)*. Salzburg, Austria: SESAR Joint Undertaking. 5

PRATS MENÉNDEZ, XAVIER, BARRADO MUXÍ, CRISTINA, VIDOSAVLJEVIC, ANDRIJA, DELAHAYE, DANIEL, NETJASOV, FEDJA, & CRNOGORAC, DUSAN. 2017. Assessing atm performance with simulation and optimisation tools: The apache project. *In: 7th SESAR Innovation Days (SIDS)*. Belgrade , Serbia: SESAR Joint Undertaking. 8

PRC. 2019 (May). *Performance Review Report: An Assessment of Air Traffic Management in Europe during the Calendar Year 2018*. Tech Report. EUROCONTROL, Performance Review Commission, Brussels, Belgium. ix, xiii, 1, 7, 8, 51, 73

PRC. 2021 (Jun). *Performance Review Report: An Assessment of Air Traffic Management in Europe*. Tech Report. EUROCONTROL, Performance Review Commission, Brussels, Belgium. 2, 7, 51

R-NEST. 2022. *Research network strategic monitoring tool (r-nest)*. https://www.eurocontrol.int/solution/rnest. Accessed: October 24, 2022. 27

REBOLLO, JUAN JOSE, & BALAKRISHNAN, HAMSA. 2014. Characterization and prediction of air traffic delays. *Transportation research part C: Emerging technologies*, **44**, 231–241. 98, 105

ROBERTS, DAVID R, BAHN, VOLKER, CIUTI, SIMONE, BOYCE, MARK S, ELITH, JANE, GUILLERA-ARROITA, GURUTZETA, HAUENSTEIN, SEVERIN, LAHOZ-MONFORT, JOSÉ J, SCHRÖDER, BORIS, THUILLER, WILFRIED, *et al.* 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, **40**(8), 913–929. 35

RUIZ, SERGIO, KADOUR, HAMID, & CHOROBA, PETER. 2019. An innovative safety-neutral slot overloading technique to improve airspace capacity utilisation. *In: 9th SESAR Innovation Days (SIDS)*. Athens , Greece: SESAR Joint Undertaking. 8, 84

SAE. 2018 (Jun). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Tech Report. Society of Automotive Engineers, Warrendale, Pennsilvània, USA. 9

SAMÀ, MARCELLA, D'ARIANO, ANDREA, & PACCIARELLI, DARIO. 2012. Optimal aircraft traffic flow management at a terminal control area during disturbances. *Procedia-social and behavioral sciences*, **54**, 460–469. 5

SANAEI, RASOUL, LAU, ALEXANDER, LINKE, FLORIAN, & GOLLNICK, VOLKER. 2019. Machine learning application in network resiliency based on capacity regulations. *In: 38th digital avionics systems conference (dasc)*. Virtual event: AIAA/IEEE. 98

SANAEI, RASOUL, PINTO, BRIAN ALPHONSE, & GOLLNICK, VOLKER. 2021. Toward atm resiliency: A deep CNN to predict number of delayed flights and ATFM delay. *Aerospace*, **8**(2), 28. 50

SCHULTZ, MICHAEL, REITMANN, STEFAN, & ALAM, SAMEER. 2021. Predictive classification and understanding of weather impact on airport performance through machine learning. *Transportation Research Part C: Emerging Technologies*, **131**, 103119. 74

SCIKIT-LEARN. 2022a. *Balanced accuracy score*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html. Accessed: December 2, 2022. 108

SCIKIT-LEARN. 2022b. *GridSearchCV*. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Accessed: December 2, 2022. 78, 108

SCIKIT-LEARN. 2022c. *Supervised learning*. https://scikit-learn.org/stable/supervised_learning.html. Accessed: December 4, 2022. 108

SESAR. 2019a (Nov). *A proposal for the future architecture of the European airspace*. Tech Report. SESAR Joint Undertaking, Brussels, Belgium. 5, 10

SESAR. 2019b (Nov). *SESAR Solution PJ.06-01 SPR-INTEROP/OSED for V3*. Tech Report. SESAR Joing Undertaking, Brussels, Belgium. 10

SESAR. 2020 (Jun). *European ATM Master Plan: Digitalizing Europe's Aviation Airspace*. Tech Report. SESAR Joint Undertaking, Brussels, Belgium. ix, 2, 9, 10

SESAR. 2022 (Jun). *European ATM Master Plan: Digitalizing Europe's Aviation Airspace*. Tech Report. SESAR Joint Undertaking, Brussels, Belgium. 9

SHETTY, KAMALA, GULDING, JOHN, KOELMAN, HARTMUT, CELIKTIN, METE, & KOELLE, RAINER. 2017 (Apr). Comparison of ATFM practices and performance in the US and Europe. *In: Integrated communications, navigation and surveillance conference (icns)*. IEEE, Herndon, Virginia, USA. 7

SILVER, DAVID, LEVER, GUY, HEESS, NICOLAS, DEGRIS, THOMAS, WIERSTRA, DAAN, & RIEDMILLER, MARTIN. 2014 (Jan). Deterministic policy gradient algorithms. *In: 31st International Conference on Machine Learning*. PMLR, Bejing, China. 22

SKYBRABY. 2022. *ATC Operations in Weather Avoidance Scenarios*. https://www.skybrary.aero/index.php/. Accessed: Novemebr 11, 2022. 77

SPATHARIS, CHRISTOS, KRAVARIS, THEOCHARIS, VOUROS, GEORGE A, BLEKAS, KONSTANTINOS, CHALKIADAKIS, GEORGIOS, GARCIA, JOSE MANUEL CORDERO, & FERNANDEZ, ESTHER CALVO. 2018. Multiagent reinforcement learning methods to resolve demand capacity balance problems. *In: 10th hellenic conference on artificial intelligence*. Patras, Peloponnesus, Greece: IEEE. 88

SPATHARIS, CHRISTOS, BASTAS, ALEVIZOS, KRAVARIS, THEOCHARIS, BLEKAS, KONSTANTINOS, VOUROS, GEORGE A, & CORDERO, JOSE MANUEL. 2021. Hierarchical multiagent reinforcement learning schemes for air traffic management. *Neural computing and applications*, 1–13. 84

STATFOR, EUROCONTROL. 2022 (june). *Forecast update 2022-2024: Recovery from covid-19 and russian invasion of ukraine*. Tech Report. EUROCONTROL, STATFOR, Brussels, Belgium. ix, 1, 2

STONE, MERVYN. 1978. Cross-validation: A review. *Statistics: A journal of theoretical and applied statistics*, **9**(1), 127–139. 34

SUNEHAG, PETER, LEVER, GUY, GRUSLYS, AUDRUNAS, CZARNECKI, WOJCIECH MARIAN, ZAMBALDI, VINICIUS, JADERBERG, MAX, LANCTOT, MARC, SONNERAT, NICOLAS, LEIBO, JOEL Z, TUYLS, KARL, *et al.* 2017. Value-Decomposition networks for cooperative multi-agent learning. *arxiv preprint*. 21

SUTTON, RICHARD S., & BARTO, ANDREW G. 1999. *Reinforcement learning: an introduction*. The MIT Press. 19

SUTTON, RICHARD S, & BARTO, ANDREW G. 2018. *Reinforcement learning: An introduction*. MIT press. 20

TAN, MING. 1993 (Aug). Multi-agent reinforcement learning: Independent vs. Cooperative Agents. *In: 10th international conference on machine learning*. 21

TANG, JIANGJUN, ALAM, SAMEER, LOKAN, CHRIS, & ABBASS, HUSSEIN A. 2012. A multi-objective approach for Dynamic Airspace Sectorization using agent based and geometric models. *Transportation research part C: Emerging technologies*, **21**(1), 89–121. 50

TIBICHTE, A, & DALICHAMPT, M. 2014 (Jan). *Atfm modelling capability*. Tech Report. EUROCONTROL, Experimental Center, Brétigny-sur-Orge, France. 7, 8, 83

TOBARUELA, GONZALO, MAJUMDAR, ARNAB, OCHIENG, WASHINGTON Y, SCHUSTER, WOLFGANG, & HENDRICKX, PETER. 2013. Enhancing cost-efficiency and reducing capacity shortages: strategic planning and dynamic shift management. *In: 10th USA/Europe Air Traffic Management Research and Development Seminar*. Chicago, IL, USA: IEEE. 3

TUMER, KAGAN, & AGOGINO, ADRIAN. 2007. Distributed agent-based air traffic flow management. *In: 6th international joint conference on Autonomous agents and multiagent systems*. Honolulu, Hawaii, USA: AAMAS. 84, 88

UHLENBECK, GEORGE E, & ORNSTEIN, LEONARD S. 1930. On the theory of the Brownian motion. *Physical Review*, **36**(5), 823. 91

WATKINS, CHRISTOPHER JCH, & DAYAN, PETER. 1992. Q-learning. *Machine learning*, **8**(3), 279–292. ix, 19, 21

WELCH, JERRY D, ANDREWS, JOHN W, MARTIN, BRIAN D, & SRIDHAR, BANAVAR. 2007. Macroscopic workload model for estimating en route sector capacity. *In: 7th USA/Europe ATM Research and Development Seminar*. Barcelona, Spain: IEEE. 5

XIE, YIBING, PONGSAKORNSATHIEN, NICHAKORN, GARDI, ALESSANDRO, & SABATINI, ROBERTO. 2021. Explanation of machine-learning solutions in air-traffic management. *Aerospace*, **8**(8), 224. 39

XU, YAN, & PRATS, XAVIER. 2018. Linear holding for airspace flow programs: A case study on delay absorption and recovery. *Ieee transactions on intelligent transportation systems*, **20**(3), 1042–1051. 50

XU, YAN, DALMAU, RAMON, MELGOSA, MARC, MONTLAUR, ADELINE, & PRATS, XAVIER. 2020. A framework for collaborative air traffic flow management minimizing costs for airspace users: Enabling trajectory options and flexible pre-tactical delay management. *Transportation research part b: Methodological*, **134**, 229–255. 98

XUE, MIN. 2009. Airspace sector redesign based on voronoi diagrams. *Journal of aerospace computing, information, and communication*, **6**(12), 624–634. 4

YU, KEMING, LU, ZUDI, & STANDER, JULIAN. 2003. Quantile regression: applications and current research areas. *Journal of the royal statistical society: Series d (the statistician)*, **52**(3), 331–350. 109