Designing Recurrent and Graph Neural Networks to Predict Airport and Air Traffic Network Delays

Sun, Junzi; Dijkstra, T.L.E.; Aristodemou, K.; Buzețelu, V.S.; Falat, T.; Hogenelst, T.G.; Prins, N.; Slijper, B.C.

Publication date
2022

Published in
10th International Conference for Research in Air Transportation

Citation (APA)
Sun, J., Dijkstra, T. L. E., Aristodemou, K., Buzețelu, V. S., Falat, T., Hogenelst, T. G., Prins, N., & Slijper, B. C. (2022). Designing Recurrent and Graph Neural Networks to Predict Airport and Air Traffic Network Delays. In D. Lovell (Ed.), *10th International Conference for Research in Air Transportation* FAA & Eurocontrol.

Designing Recurrent and Graph Neural Networks to Predict Airport and Air Traffic Network Delays

Junzi Sun*, Tristan Dijkstra, Constantinos Aristodemou,
Vlad Buzetelu, Theo Falat, Tim Hogenelst, Niels Prins, Benjamin Slijper

Faculty of Aerospace Engineering,
Delft University of Technology,
Delft, the Netherlands
*corresponding: j.sun-1@tudelft.nl

Abstract—**In this paper, we propose open machine learning models that can provide airport delay predictions in a network with an error of around or less than five minutes. Due to the complexity of different components of air traffic networks, traditional flight performance model-based predictions fall short when dealing with numerous flights and often are not able to deal with delays that propagate among airports in a network. In this study, we employ three different machine learning models to predict delays at three different scopes: individual flights, airports, and the network of airports. Consequently, we tested three approaches with different levels of complexity, including statistical regression models, recurrent neural networks, and spatial-temporal graph attention neural networks. We conduct experiments for all three types of models using the Eurocontrol research data archive. After training and testing with two years of data covering the top 50 European airports, our models produce prediction errors of around or less than 5 minutes with look-ahead time up to 3 hours. These metrics have shown a significant advancement compared to existing prediction models. We also openly share this model to support open science in aviation.**

***Keywords* —** Flight delay, airport delay propagation, random forest, recurrent neural network, graph attention neural network

I. Introduction

Flight delays significantly impact all stakeholders in the air traffic system, including passengers, airlines, airports, and air navigation service providers. A previous study has shown that around 20% of flights in Europe experienced delays longer than 5 minutes in the year 2018 [1]. Delays can be either planned or unplanned. Planned delays include air traffic flow management (ATFM) delays before an aircraft takes off, and en-route ATFM delays. Delays can also be categorized as reactionary or primary. Reactionary delays are caused by the late arrival of the previous journey of an aircraft, while primary delays are delays caused by other reasons.

From an airport perspective, the overall delays among all arriving and departing flights are two key indicators that evaluate the airport's performance. Rather than focus on delays of individual flights, the average delay of all flights can also further the understanding of the propagation of delays among the networks of airports [2].

Over the past year, several studies have focused on the prediction of delays. Some research has proposed a classification approach, which essentially predicts the severity of delays using data-driven classifiers [3], [4]. More accurate regression models have also been proposed. For example, study [2] developed a non-parametric statistical approach to model daily and seasonal trends and uses a mixture distribution to estimate residual errors. A random forest model was proposed by [5] and achieved a median estimation error of around 20 minutes. Another study [6] proposed several methods including the Markov Jump Linear System, regression trees, and a deep neural network that achieved an airport estimation error of around 7 minutes for a 2-hour prediction horizon for the top 30 airports in the USA. In a more recent study [7], a recurrent neural network architecture was employed that embeds the graph information to predict airport delays. When this model is training on a non-sampled dataset, the method provides a mean error of around 6.5 minutes for US airports for short look-ahead time.

Traffic prediction is not a problem unique to air transport; recent research dealing with ground transport delay [8] employed a more advanced graph attention neural network, to estimate road traffic. The graph attention network was found to be superior in considering features from connection transportation nodes. However, in [8], connections between nodes are represented by constant matrices, which is not the case for air traffic networks.

In this paper, we address the air traffic delay problem at three different levels and propose three separate models that are better suited to tackle each regression problem. For flight and airport delay predictions, we design a classical machine learning approach and recurrent neural network approaches. Both the classical random forest approach and recurrent neural network approach provide estimation errors that are comparable to or better than the state-of-the-art models.

The main innovation of this paper is the development of dynamic spatial-temporal graph attention (DST-GAT) neural networks that can be employed to accurately model and estimate the delays in the network of airports. It allows us to achieve a very low mean estimation error across the top 50 airports in Europe. It not only improves the estimation accuracy of current prediction models but also provides a new approach to modeling and studying delay propagation in the air transport network. In the experiment section of this paper, we conduct a thorough analysis of the error using two years of data from the Eurocontrol R&D data archive.

Furthermore, the source code of all our models is publicly shared along with the publication of this paper to encourage more open science in aviation. Currently, this is one of few delay prediction models that are publicly shared in aviation research.

II. Methodology

A. Data exploration

In this study, we use flights from the EUROCONTROL R&D Data Archive [1] to model and evaluate various delay predictions. This dataset provides all commercial flights that are registered on the EUROCONTROL network. The flight plan and flown trajectory are provided for each flight. It is worth noting that only four months (March, June, September, and December) of data each year are provided in this dataset, and hence these are the only available months for constructing the prediction model.

The two predictor key parameters we consider are departure delays and arrival delays. They are both calculated based on the flight plan and

[1] https://www.eurocontrol.int/dashboard/rnd-data-archive

flown trajectory. The filed and actual off block time (FOBT and AOBT) are already calculated and provided for all the flights each month. The associated airports (departure and arrival) are also presented in the dataset.

An additional feature, capacity at an airport, is defined as the fraction of total arriving and departing flights compared to the maximum capacity of the airport. Temporal features, month, and day of the week are also included. For the single airport and network of airport models, individual flight data is then aggregated over the same time intervals. An overview of the data flow can be seen in Fig. 1.

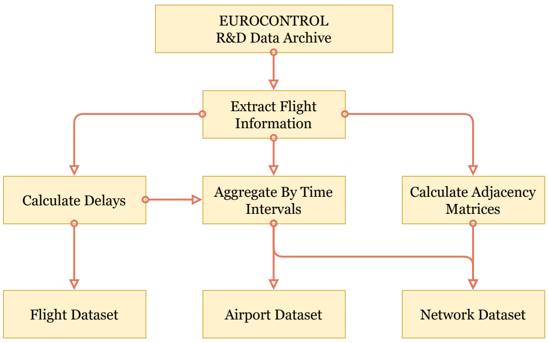
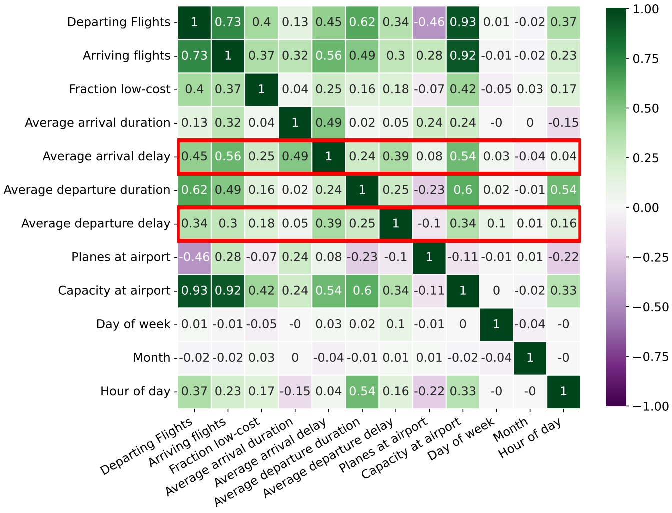


Fig. 1: Data extraction flow for all three models

With the basic parameters generated from the dataset, we perform an exploratory analysis of features and predictors (average arrival and departure delays of an airport). Different levels of correlations can already be seen between the features. Fig. 2 illustrates such correlations. These correlations form the basis for our airport and network delay predictions models.



	Departing Flights	Arriving flights	Fraction low-cost	Average arrival duration	Average arrival delay	Average departure duration	Average departure delay	Planes at airport	Capacity at airport	Day of week	Month	Hour of day
Departing Flights	1	0.73	0.4	0.13	0.45	0.62	0.34	-0.46	0.93	0.01	-0.02	0.37
Arriving flights	0.73	1	0.37	0.32	0.56	0.49	0.3	0.28	0.92	-0.01	-0.02	0.23
Fraction low-cost	0.4	0.37	1	0.04	0.25	0.16	0.18	-0.07	0.42	-0.05	0.03	0.17
Average arrival duration	0.13	0.32	0.04	1	0.49	0.02	0.05	0.24	0.24	-0	0	-0.15
Average arrival delay	0.45	0.56	0.25	0.49	1	0.24	0.39	0.08	0.54	0.03	-0.04	0.04
Average departure duration	0.62	0.49	0.16	0.02	0.24	1	0.25	-0.23	0.6	0.02	-0.01	0.54
Average departure delay	0.34	0.3	0.18	0.05	0.39	0.25	1	-0.1	0.34	0.1	0.01	0.16
Planes at airport	-0.46	0.28	-0.07	0.24	0.08	-0.23	-0.1	1	-0.11	-0.01	0.01	-0.22
Capacity at airport	0.93	0.92	0.42	0.24	0.54	0.6	0.34	-0.11	1	0	-0.02	0.33
Day of week	0.01	-0.01	-0.05	-0	0.03	0.02	0.1	-0.01	0	1	-0.04	-0
Month	-0.02	-0.02	0.03	0	-0.04	-0.01	0.01	0.01	-0.02	-0.04	1	-0
Hour of day	0.37	0.23	0.17	-0.15	0.04	0.54	0.16	-0.22	0.33	-0	-0	1

Fig. 2: Correlation heatmap for features used in airport and network delay models, built with a subset of data for EGLL on January, 1, 2019

B. Flight arrival delay predictions

The flight arrival delay model focuses on predicting the single-event arrival delay for individual flights. It also serves as the basis for the later airport and network delay models. Compared to the later models, three flight-specific parameters are available and used for training the prediction models, which are filed off-block time, filed arrival time, and departure delay at the origin airport.

Table I: Features for predicting flight arrival delays

Features	Type
Departure delay	Numerical (minutes)
Departure airport	Categorical
Filed off-block time	Numerical (15-min interval)
Filed arrival time	Numerical (15-min interval)
Aircraft operator	Categorical
Month	Categorical
Day of week	Categorical
Airport capacity	Numerical (percentage)

Several common machine regression models are constructed for flight arrival delay prediction. These models are K-Nearest Neighbor (KNN) [9], Support Vector Machine (SVM) [10], Multiple Linear Regression (MLR), and Random Forest (RF) [11].

We use cross-validations and grid searches to obtain the best hyperparameters for the machine learning model. Once the models are trained, we evaluate and compare the performance of all four different models using three error metrics, including mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2).

C. Airport delay predictions

The airport delay model aims to predict average arrival and departure delays for a given airport. Our main objective is to effectively capture the dynamic of delay evolution during daily operations. To this extent, we propose a recurrent neural network, specifically, the long-short term memory (LSTM) network [12] to consider the spatial dynamic of airport delays. The LSTM network can model time-series features using the combination of input, forget, and output gates, where past states and current states are adapted and used to predict future states.

Features in Table II are used for the LSTM network. It is worth noting that each feature also has an additional dimension with several time steps representing information from the past for predicting future delays at the airport.

Table II: Features for predicting airport arrival and departure delays

Features (from the past)	Type
Departing flights	Numerical
Arriving flights	Numerical
Fraction of low-cost flights	Numerical
Mean arrival flight duration	Numerical (minutes)
Mean departure flight duration	Numerical (minutes)
Average arrival delay	Numerical (minutes)
Average departure delay	Numerical (minutes)
Filled capacity at the airport	Numerical (percentage)
Airplanes at the airport	Numerical
Month	Categorical
Day of week	Categorical
Hour of the day	Categorical

For all look-back time steps, all features in the previous table are constructed as input data. It is worth noting that past delay forms the basis of the LSTM model, which is combined with other features. The predictors are arrival and departure delays for all look-ahead time steps.

Fig. 3 shows the structure of the neural network, where two LSTM layers are followed by several dense layers, and finally reshaped into the correct output dimensions.

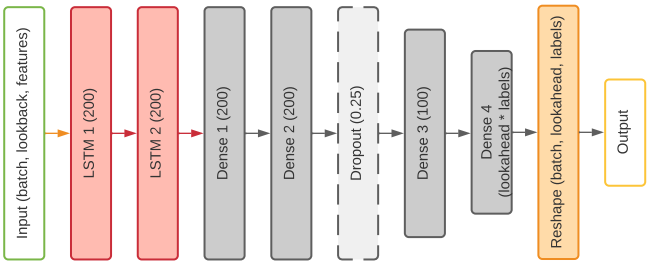

Input (batch, lookback, features)
LSTM 1 (200)
LSTM 2 (200)
Dense 1 (200)
Dense 2 (200)
Dropout (0.25)
Dense 3 (100)
Dense 4 (lookahead * labels)
Reshape (batch, lookahead, labels)
Output

Fig. 3: Neural network structure for airport average arrival and departure delay estimation

D. Delay predictions in the network of airports

1) Spatial-temporal graph model: The model in this section is designed for predicting delays at a network of airports and studying the propagation of delays among these airports. We need to consider both the time-varying features of an individual airport and also its relationships with connected airports. The network of airports is represented by a graph model. Each airport in the network acts as a node with its feature state for every time step. The edges of the graph represent the relationships between the airports.

To this extent, we propose a dynamic spatial-temporal graph attention neural network (DST-GAT) to model and predict both arrival and departure delays of all airports in the network.

Such a graph attention neural network can efficiently consider the delay, capacity, and aggregated flight information from all connected airports. We also propose a specific type of network that allows the properties of graph edges to be updated dynamically over time. As the result, without the need for a propagation model, as proposed in [13], the dynamic graph neural network can more effectively capture the propagation.

2) Adjacency matrices: Adjacency matrices are used to describe specific relationships between all connected nodes. In our case, two matrices are designed. The first adjacency matrix A_{dist}, proposed according to study [7], describes the geographical closeness of two airports. We intend to capture the correlations of airspace situations for airports that are located closer to each other. Equation 1 is used to define the element of this static matrix:

$$a_{ij} = \begin{cases} \exp\left(\dfrac{\mathsf{dist}^2(p_i, p_j)}{\sigma^2}\right) & \text{if } \mathsf{dist}(p_i, p_j))^2 < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where p represents the position of an airport and σ is the standard deviation of all distances among airport pairs. The threshold ϵ is set to 400km empirically.

The second adjacency matrix, A_{flight}, models the number of flights between two airports at each time interval. This matrix is dynamically updated for each time step between all airport pairs.

These matrices are then summed with different weights, shown by Equation 2:

$$A_t = \alpha A_{\mathsf{dist}} + (1 - \alpha) A_{\mathsf{flight,t}} \tag{2}$$

where α is the weighting factor. It is selected to be 0.4 empirically to reflect the higher importance of connecting flights than geographical distances.

E. Graph Attention Networks

Over the past years, several types of graph neural networks have been proposed in the computer science domain. Since the delay propagates among the airports in an air traffic network, features from neighboring nodes should be a significant factor in the prediction. We chose to adopt the Graph Attention Network (GAT) [14] as our graph network layer. The GAT architecture allows us to efficiently exploit relevant information from adjacent airports. Fig. 4 shows the update of a single node in the graph network based on the states and relationships of neighbor states.

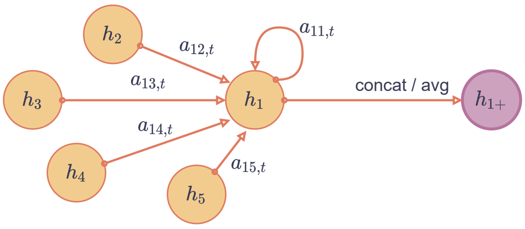


Fig. 4: Illustration of GAT neighborhood attention for a single node. Node 1 obtains attention coefficient from all neighbors and computes the linear combination of their features. After that, the next level feature of node 1 (h_{1+}) is aggregated.

The advantage of using GAT over more common Graph Convolutional Networks (GCN) is that the importance of connected airports is updated dynamically. In addition, edge features can also be more easily exploited by the GAT network.

In our design, we also want to model the time dynamic of the propagated delays. Hence, the GAT layer is followed by two LSTM layers. The principle and structure of this LSTM layer are similar to the one used for airport-level delay prediction. It performs future predictions based on features from look-back steps. In the final DST-GAT architecture, for every sequence of predictions, the same number of look-back steps are processed by the GAT layer and then the LSTM layers. The structure of the model is illustrated in Fig. 5.

The same features from Table II are used for training our DST-GAT model.

III. Experiments and Results

To thoroughly validate the performance of all three types of prediction models at different levels, we downloaded three years of data from 2017 to 2019 from the Eurocontrol R&D data archive. In total, this subset of data contains all Eurocontrol flights that occurred in March, June, September, and December each year.

To reduce the impact of extreme outliers, we apply a filter to remove flights of extreme values that correspond to delays larger than 90 minutes or smaller than -30 minutes (early arrivals or departures). The removed flights account for approximately 0.7% of all flights.

For the airport- and network-level delay predictions, in the following experiments, the proposed neural networks model and predict the average arrival and departure delays with a time window of 30 minutes. It is worth noting that this time window can be adapted based on targets of different use cases.

A. Flight arrival delay prediction

To evaluate the single arrival delay event prediction from the random forest model, we make use of all flights between 2017 and 2019, arriving at the top 25 European airports. For each airport, a separate random forest model is constructed. The estimation errors of all flights are illustrated in Fig. 6, organized by the airports.

Table III shows the random forest estimator performance metrics across all airports. In general, the MAE is found to be between 3.8 and 7.7 minutes, with a mean error of around 4.5 minutes across all airports. The RMSE is found to be between 5.1 and 10.5 minutes, with a mean of 6.3 minutes across all airports. The R^{2}score describes how well a model performs compares to the simple mean, which is in general around 0.8. It indicates that the models can model a great part of the variance shown in arrival flight delays.

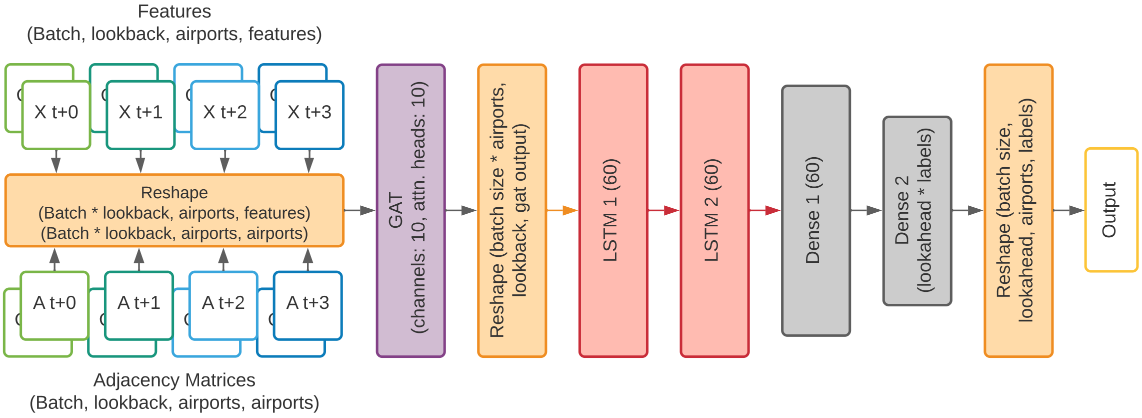

Features
(Batch, lookback, airports, features)
X t+0
X t+1
X t+2
X t+3
Reshape
(Batch * lookback, airports, features)
(Batch * lookback, airports, airports)
A t+0
A t+1
A t+2
A t+3
Adjacency Matrices
(Batch, lookback, airports, airports)
GAT
(channels: 10, attn. heads: 10)
Reshape (batch size * airports,
lookback, gat output)
LSTM 1 (60)
LSTM 2 (60)
Dense 1 (60)
Dense 2
(lookahead * labels)
Reshape (batch size,
lookahead, airports, labels)
Output

Fig. 5: Model structure of DST-GAT neural network used for delay predictions in a network of airports

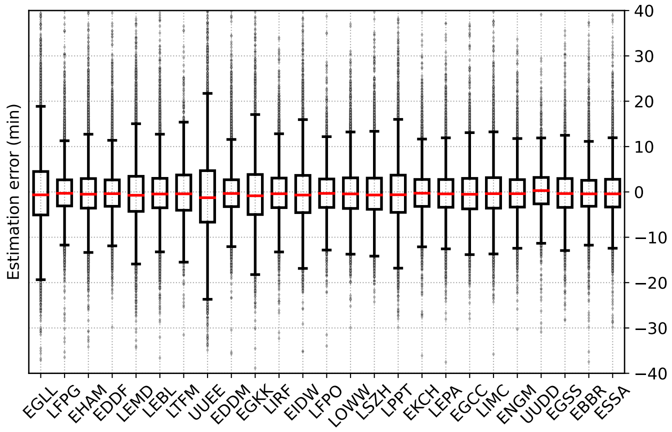


Fig. 6: Estimation error of arrival delays at the top 25 airports in Europe from 2017 to 2019

Table III: Random forest: error metrics for all flight arriving at top European airports

	EGLL	LFPG	EHAM	EDDF	LEMD	LEBL	LTFM	UUEE	EDDM	EGKK
MAE	6.02	3.77	4.32	3.83	5.19	4.34	5.07	7.67	3.83	5.68
RMSE	7.90	5.14	5.93	5.25	7.09	6.03	6.96	10.45	5.24	7.55
R^2	0.75	0.80	0.85	0.83	0.71	0.81	0.74	0.58	0.81	0.77

	LIRF	EIDW	LFPO	LOWW	LSZH	LPPT	EKCH	LEPA	EGCC	LIMC
MAE	4.24	5.38	4.08	4.30	4.48	5.42	3.82	4.11	4.51	4.51
RMSE	5.75	7.31	5.58	5.75	6.01	7.43	5.15	5.79	6.26	6.31
R^2	0.78	0.74	0.80	0.78	0.78	0.76	0.79	0.83	0.76	0.80

B. Airport-level arrival and departure delay prediction using LSTM networks

To demonstrate the performance of airport delay predictions, we selected London Heathrow Airport (EGLL) as the example, which is one of the busiest airports in Europe. 80% of the data is used for training, while 20% of the data is used for testing. Specifically, flight data associated with EGLL from January 1, 2018, to September 12, 2019, are used to train the model. The remaining date in 2019 (from September 13 to December 31) is used as the testing dataset to examine the performance of models.

It is important to emphasize that, compared to the commonly used random split of training and testing datasets, we make such a strict split by a cutoff date to be rigorous for the validation process. This way, we eliminate any potential data leak between training and test data. Thus, the results should indicate the performance of the model in the least optimal situation, i.e. the model trained up to September is still able to be used for predictions in December 2019.

Table IV shows the performance of our LSTM network trained and tested with the EGLL dataset. With different look-ahead times, the MAE for arrivals is between 4.6 minutes and 5.4 minutes, while the departure delay errors are between 2.2 and 2.3 minutes.

Table IV: LSTM: arrival and departure delay metrics, EGLL

(a) Arrival delay

look-ahead	MAE	RMSE	R^2
30 min	4.59	6.31	0.69
60 min	4.59	6.43	0.68
90 min	4.78	6.73	0.65
120 min	4.91	7.02	0.62
150 min	5.24	7.49	0.56
180 min	5.43	7.73	0.53

(b) Departure delay

look-ahead	MAE	RMSE	R^2
30 min	2.20	3.29	0.45
60 min	2.18	3.36	0.43
90 min	2.27	3.54	0.37
120 min	2.30	3.55	0.37
150 min	2.24	3.50	0.38
180 min	2.30	3.57	0.36

Fig. 7 shows the difference between actual and estimated delays for a number of days in the testing dataset. For better illustration, 20 out of the 50 days in the test dataset are plotted. By comparing the prediction and actual delays, we see a great level of prediction accuracy that captures the delay dynamic.

In terms of computational performance, for such a signal airport model with approximately 8 months of data, the training takes about one minute on an Nvidia Quadro K620 GPU (from the year 2014).

C. Network-level arrival and departure delay prediction using DST-GAT graph neural networks

The final experiment is designed to test the performance of the spatial-temporal graph attention networks at the large network level. In this test, we selected the top 50 airports in Europe to demonstrate the performance of our DST-GAT model. The training and testing datasets are constructed similarly to the previous experiments (80% for training, 20% for testing). Data from January 1, 2018, to September 12, 2019,

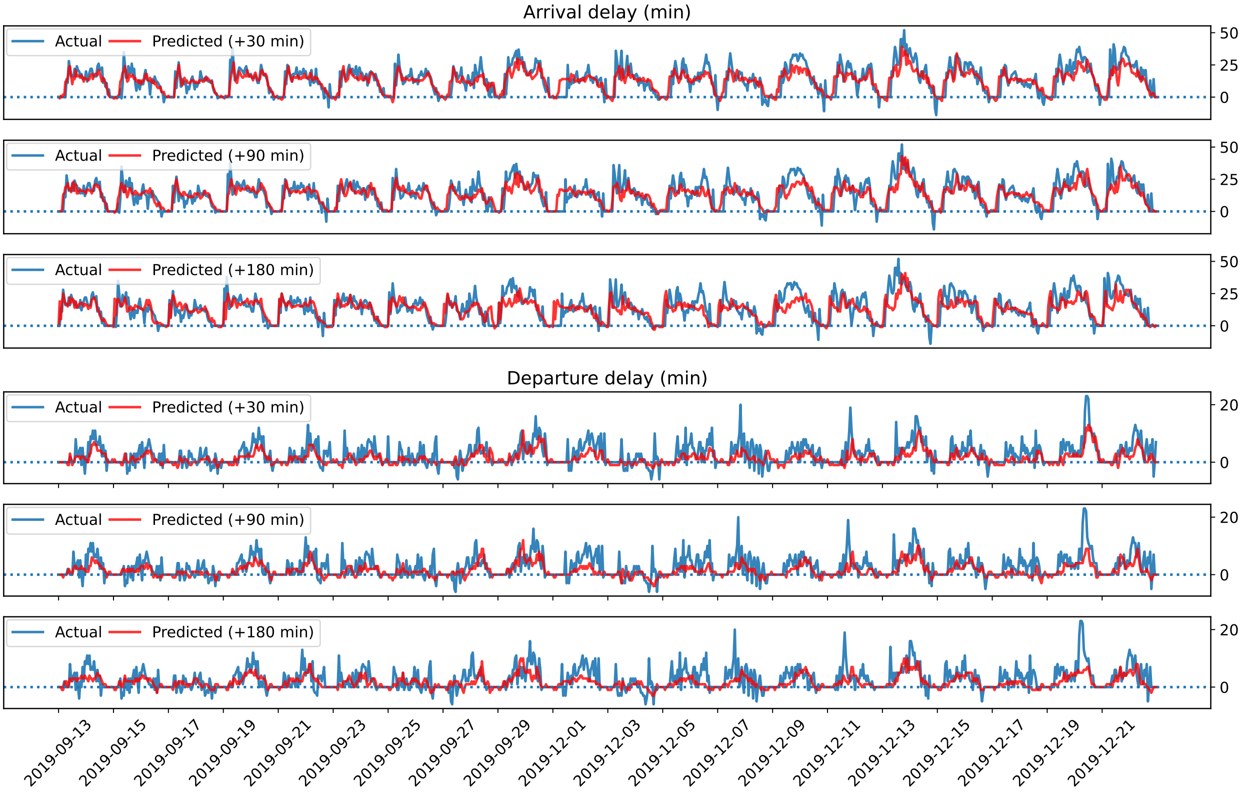


Fig. 7: LSTM: Average arrival and departure delay with different look-ahead times for EGLL

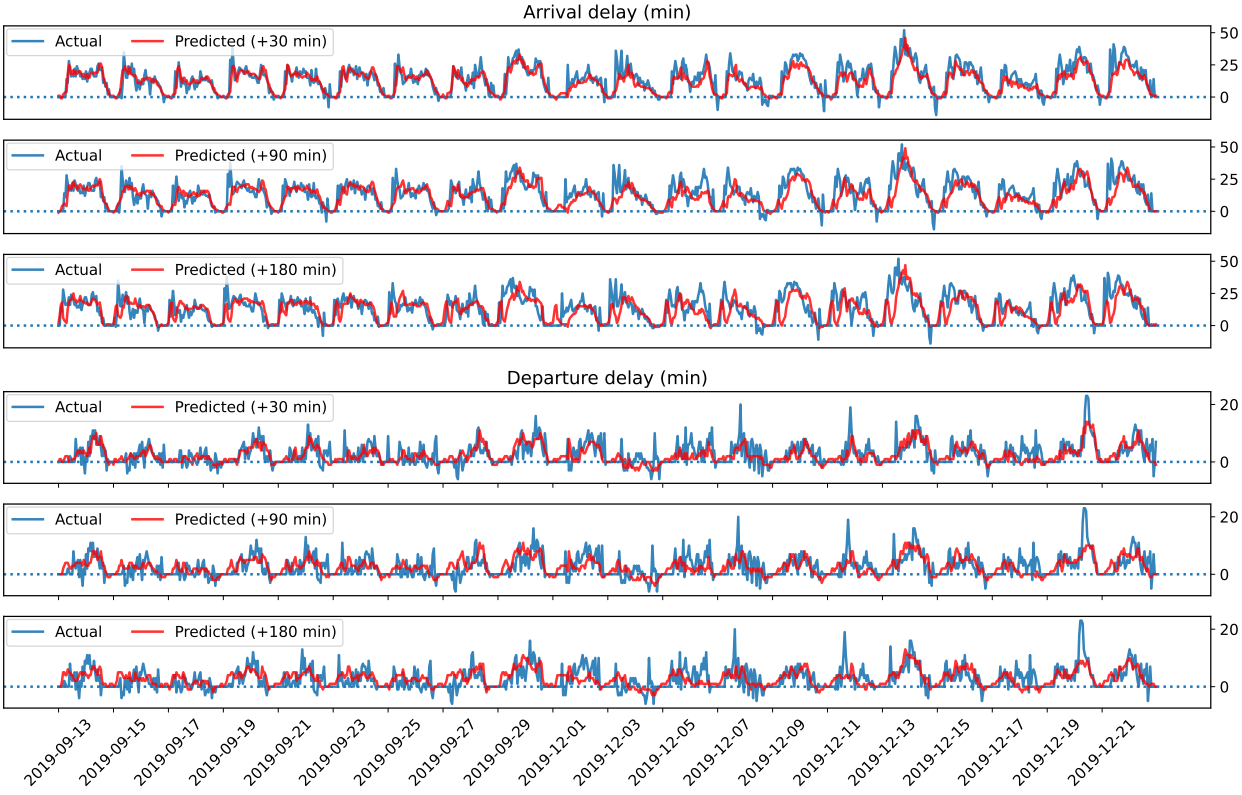


Fig. 8: DST-GAT: average arrival and departure delay with different look-ahead times (showing EGLL as an example).

is used for model training, and data from September 13 to December 31 is used for testing. Features from Table II are considered by the model.

Due to the complexity of the GAT network and the higher quantity of data, the training of the entire model takes about 40 minutes for the network of 50 airports (on the same Quadro K620 GPU from 2014). Compared to the previous single airport LSTM network, the computational complexity increases linearly.

Table V shows the error metrics for all airport arrival and departure delays in the testing dataset. The MAE for arrival delay prediction is around or less than 5 minutes, while departure delay error is around or less than 4 minutes. The visualization of the testing and predicted delays are shown in Fig. 8

Table V: DST-GAT: arrival and departure delay metrics for all top 50 airports

(a) Arrival delay

look-ahead	MAE	RMSE	R^2
30 min	4.60	7.09	0.38
60 min	4.75	7.27	0.35
90 min	4.86	7.41	0.32
120 min	4.93	7.51	0.30
150 min	4.98	7.58	0.29
180 min	5.01	7.63	0.28

(b) Departure delay

look-ahead	MAE	RMSE	R^2
30 min	3.68	5.87	0.35
60 min	3.79	6.01	0.32
90 min	3.89	6.14	0.29
120 min	3.96	6.24	0.27
150 min	4.00	6.29	0.25
180 min	4.04	6.35	0.24

The DST-GAT model can also provide delay estimation at individual airports, similar to the previous airport-level LSTM model. Table VI shows a similar level of accuracy as the previous LSTM model that is specifically trained for EGLL.

Table VI: DST-GAT: arrival and departure delay metrics for EGLL only

(a) Arrival delay

look-ahead	MAE	RMSE	R^2
30 min	4.33	6.28	0.69
60 min	4.70	6.76	0.64
90 min	5.16	7.47	0.56
120 min	5.54	8.06	0.49
150 min	5.83	8.38	0.45
180 min	5.99	8.56	0.43

(b) Departure delay

look-ahead	MAE	RMSE	R^2
30 min	2.22	3.15	0.50
60 min	2.38	3.30	0.45
90 min	2.51	3.46	0.39
120 min	2.52	3.51	0.37
150 min	2.52	3.54	0.36
180 min	2.60	3.61	0.34

Furthermore, the DST-GAT model has a superior capability in predicting delay evolution at the network level. Fig. 9 shows the evolution of arrival delay predicted with 60-minutes of look-ahead time during the day at four different hours. The size of the circle represents the delays, i.e., larger circles represent larger delays. The color represents the absolute estimation error, i.e., the darker colors represent larger estimation errors.

The estimation performances also vary depending on the airports. In Fig. 10, we show the MAE and RMSE of arrival delay estimation for all airports. The selected look-ahead is 60 minutes. We find that the MAE varies between 1.7 minutes and 6.4 minutes, with an average value of around 4.7 minutes. When examining the RMSE, the range is between 3.4 and 10.5 minutes, with an average value of 7.2 minutes. Both sets of error metrics show a good level of accuracy across the entire network.

Fig. 11 shows the error metrics for departure delay estimation errors. The MAE range is between 1.3 and 7.6 minutes, with an average value of 3.8 minutes. The RMSE is between 2.2 and 10.9 minutes with an average of 5.8 minutes. Both sets of errors are smaller than the arrival delay prediction errors shown in the previous figure.

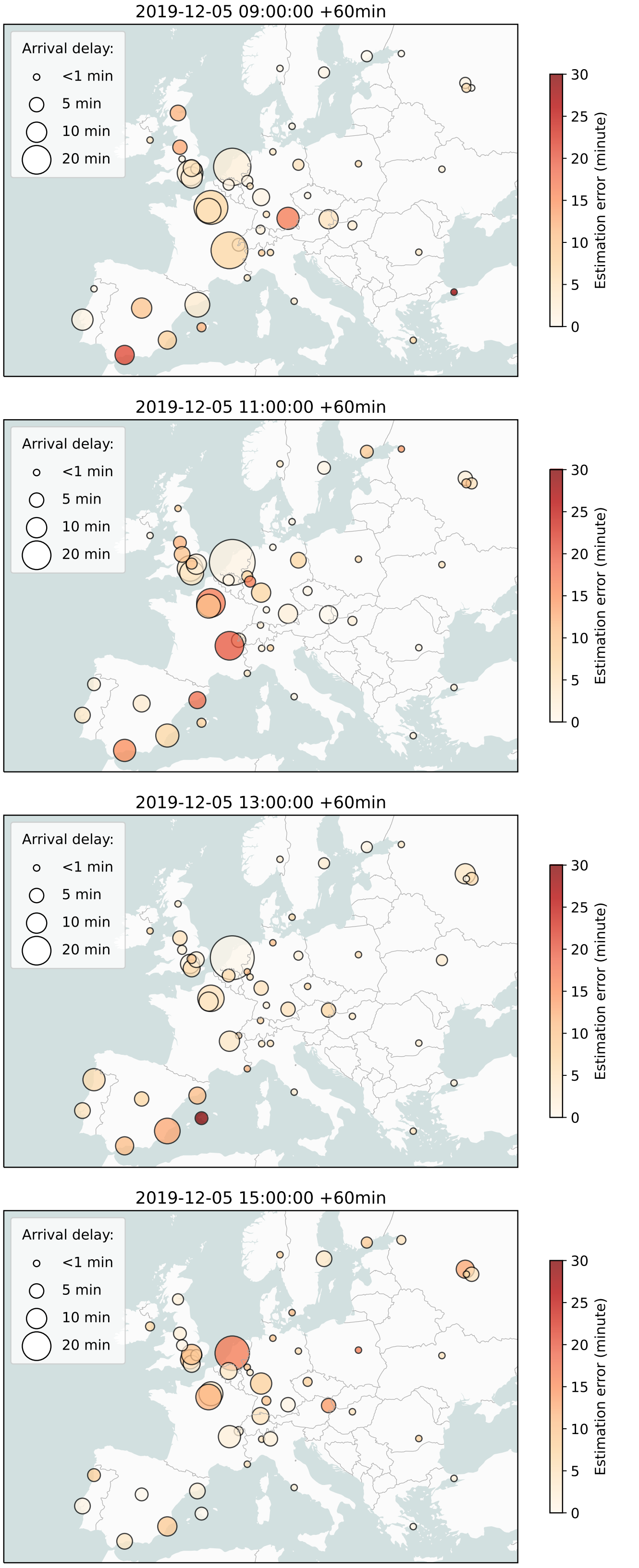

2019-12-05 09:00:00 +60min
2019-12-05 11:00:00 +60min
2019-12-05 13:00:00 +60min
2019-12-05 15:00:00 +60min
Arrival delay:
<1 min
5 min
10 min
20 min
Estimation error (minute)
0
5
10
15
20
25
30

Fig. 9: Evolution of arrival delay predictions among top airports in the European network

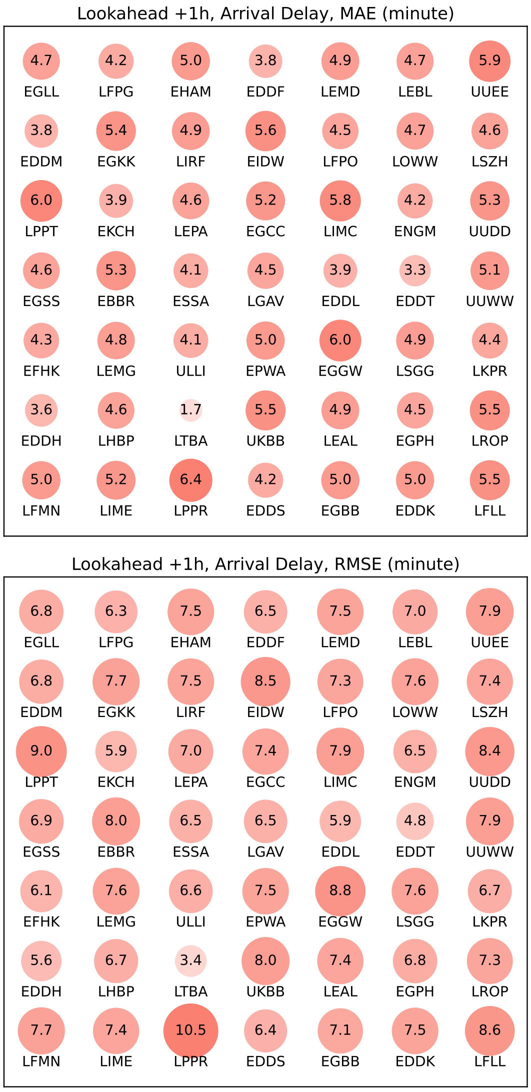


Fig. 10: Arrival delay errors for top European airports

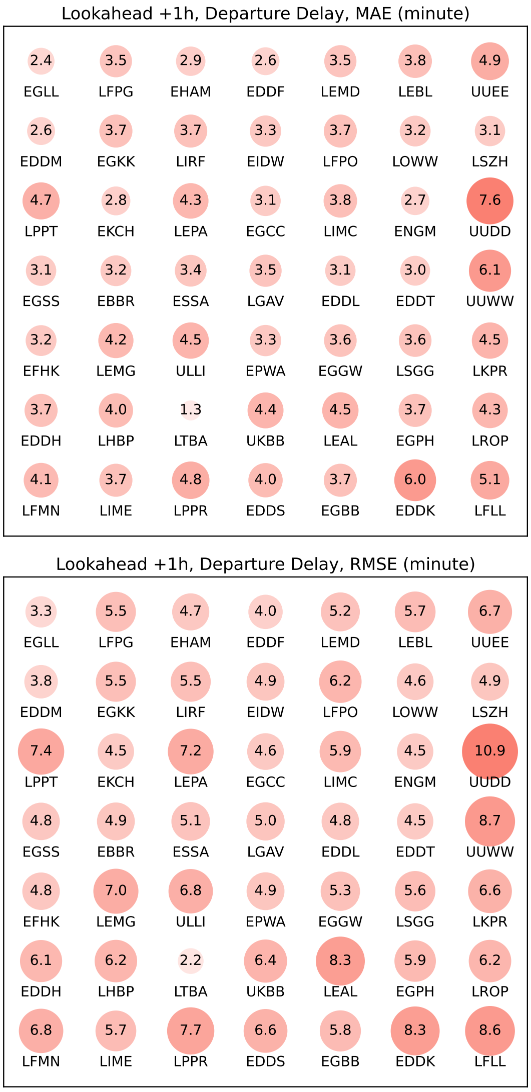


Fig. 11: Departure delay errors for top European airports

IV. Discussion

In this session, we reflect on our design choices and experiment processes. We also give more insight into the models and their performance.

A. Why so many models?

The main reason for the flight delay prediction model is to establish a baseline for our last two models. For such a model, we essentially need to design a regressor with a single-output (delay), based on inputs (shown in Table I) that contain both numerical and categorical features. It is also known that flight delays have a strong correlation with the date and time [2]. Hence, the best algorithm should be able to efficiently handle these features. That is why, among the four models that are tested, the random forest yields the best results. Since our main focus is on airport and network delays, we did not continue with other models. However, we believe other methods, like Gradient Boosting Machines [15], would potentially provide similar or better estimation performance.

One of the main challenges in this research is to match the complexity of the problem with the complexity of the model and the data that is available to construct the model. The final DST-GAT generally outperforms the single airport LSTM model in terms of lower prediction error. However, it requires a complete overview of data and a longer training time. This is often not practical for operations at a single airport. Hence, for single airport delay prediction, we still recommend the use of a simpler LSTM model with data that is available for that particular airport in Table II. By constructing separate models for the different airports, the LSTM is also likely to capture local variance that is specific to the airport.

B. The secret of model parameters

For the regression models, we make use of the grid search cross-validation approach to identify the best hyperparameters. This can be performed when the machine learning model is not too complex.

For the LSTM model, the main challenge is to figure out the best architecture, including the layer design and the number of neurons in each layer. After numerous trials and tests, we settled on the current LSTM architecture. Regarding the network design, we leave enough margin of complexity in the layers and allow the model to discover the correlations, hence the two LSTM layers with 200 neurons and several following dense layers.

We also need to balance the number of look-back time steps and look-ahead time steps to be considered in the prediction model, that is, how many hours of the old data we want to use for predicting the future delays. We settled on three hours of past data for up to three hours of future predictions in the paper. This value can be easily adapted for different use cases.

The choice of model parameters in the DST-GAT model is almost entirely constrained by GPU performance. In this study, we only have access to a fairly old GPU with only 2GB of memory. The choice of the number of nodes, number of attention heads, number of channels, and floating number computation precision are balanced to fit the entire training dataset on the GPU. All variables can be found in our shared code. A better GPU would bring both speed and high accuracy to the model.

C. Our path to the DST-GAT model

Our model is strongly inspired by two previous studies mentioned in the introduction [7] and [8]. The first paper proposed a graph-based convolutional neural network approach for airport delay, while the second paper proposed a static graph attention network for ground transport. The first model has the limited ability to consider dynamic features from neighboring airports, while the second paper relies on the static adjacency matrices, which are not ideal for the air transport network.

Based on these state-of-the-art research works, we propose to apply the dynamic adjacency matrices to the graph attention network, which is further enhanced by recurrent layers proposed in [7]. Hence, the model is considered a dynamic spatial-temporal graph attention network.

The most complex part of this research is to design the data model that can capture features representing interactions between airports in the proposed DST-GAT architecture. After numerous hours of design and testing, we found the right formula for arranging the data in the right format. These models are also shared in our open code.

D. Error metrics

In most error metrics tables, we can observe lower departure delay prediction errors than arrival delay prediction errors. This does not translate into a better performance of departure delay prediction. On the contrary, the prediction performance is worse for departures, judging by the lower coefficient of determination (R^2score). The lower R^2score suggests larger variances exist in departure delays, which are not captured by the model. We can also observe this in Figures 7 and 8, where a larger high frequency variations in departure delays cannot be captured by our models.

We think such phenomena are expected since there are other factors we are currently not able to model, including, for example, regulations for air traffic flow management delays, uncertainty in airport operations, and weather conditions. These factors would be the key to improving the prediction models in the future.

V. Conclusion

In this paper, we proposed three types of machine learning models to tackle the delay prediction problem in air traffic management research. We started with the simplest form of flight arrival delay prediction and continued on the airport-level delay prediction. Finally, we explored a new approach to model and predict delays at the network level. We selected a random forest model for single event flight arrival delay prediction as a baseline. For airport and network delays, we proposed a long short-term memory (LSTM) architecture and a dynamic spatial-temporal graph attention network (DST-GAT) architecture. Both LSTM and DST-GAT produced a good level of prediction performance with a mean absolute error of around or lower than 5 minutes in general, for both arrival and departure delays. We believe this is among the most accurate estimations for air traffic delays.

The DST-GAT architecture not only predicts delays for the airport in the network, but it can also study and predict the propagation of delays in the entire air traffic network. Currently, we consider only the distance and number of flights as features for the edges of the graph model, the model can be further extended to include other factors such as regulations of air traffic flow management, weather, and restrictions of airspace, which would bring a future improvement in both departure and arrival delays. The current adjacency matrices in DST-SAT can also be further optimized, including the threshold of the distance matrix and the weighting factor that combines both matrices.

In this paper, we used the Keras and Spektral [16] libraries as the base for our experiments. However, by the end of the project, we notice PyTorch may have better support for this specific problem with PyTorch Geometric and Geometric Temporal packages. It is also likely to bring better performance in training and modeling. We recommend that future research also consider that as an alternative to our approaches.

During the literature survey, it was hard for us to reproduce previous airport delay models, due to a lack of insight into their designs. In the spirit of open access, we shared the source code of the model along with the publication of this paper.[2] We hope this will assist future researchers to reuse our results.

Acknowledgment

We would like to thank Tom Viering for the valuable discussion on neural networks and for his feedback on the paper.

References

[1] L. Carvalho, A. Sternberg, L. Maia Goncalves, A. Beatriz Cruz, J. A. Soares, D. Brandão, D. Carvalho, and E. Ogasawara, "On the relevance of data science for flight delay research: a systematic review," *Transport Reviews*, vol. 41, no. 4, pp. 499–528, 2021.

[2] Y. Tu, M. O. Ball, and W. S. Jank, "Estimating flight departure delay distributions—a statistical approach with long-term trend and short-term pattern," *Journal of the American Statistical Association*, vol. 103, no. 481, pp. 112–125, 2008.

[3] N. Kuhn and N. Jamadagni, "Application of machine learning algorithms to predict flight arrival delays," *CS229*, 2017.

[4] J. Qu, T. Zhao, M. Ye, J. Li, and C. Liu, "Flight delay prediction using deep convolutional neural network based on fusion of meteorological data," *Neural Processing Letters*, vol. 52, no. 2, pp. 1461–1484, 2020.

[5] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation research part C: Emerging technologies*, vol. 44, pp. 231–241, 2014.

[6] K. Gopalakrishnan and H. Balakrishnan, "A comparative analysis of models for predicting delays in air traffic networks," ATM Seminar, 2017.

[7] W. Zeng, J. Li, Z. Quan, and X. Lu, "A deep graph-embedded lstm neural network approach for airport delay prediction," *Journal of Advanced Transportation*, vol. 2021, 2021.

[8] C. Zhang, J. James, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

[9] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] L. Pastorino and M. Zanin, "Air delay propagation patterns in europe from 2015 to 2018: an information processing perspective," *Journal of Physics: Complexity*, vol. 3, no. 1, p. 015001, 2021.

[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[15] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[16] D. Grattarola and C. Alippi, "Graph neural networks in tensorflow and keras with spektral [application notes]," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 99–106, 2021.

[2] https://github.com/junzis/atdelay