A Transformer-based Trajectory Prediction Model to Support Air Traffic Demand Forecasting

Vos, R.W.; Sun, Junzi; Hoekstra, J.M.

Publication date
2024

Document Version
Final published version

Published in
Proceedings International Conference on Research in Air Transportation

A Transformer-based Trajectory Prediction Model to Support Air Traffic Demand Forecasting

Reinier Vos, Junzi Sun*, and Jacco Hoekstra
Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands
*corresponding email: j.sun-1@tudelft.nl

***Abstract*—Air traffic sector demand and capacity balancing are necessary for safe and efficient flight execution. Demand and capacity are determined in current operations based on schedules and flight plans. This research aims to improve air traffic demand forecasting by exploring machine learning-based trajectory prediction, specifically the newly emerged transformer-based neural network models. The predicted trajectories are considered to improve demand forecasts for Air Traffic Control in the Netherlands. We successfully built a transformer neural network using available traffic messages from the EuroControl B2B connection and actual trajectories obtained from the OpenSky ADS-B repository. A new lost function is specifically designed to improve this prediction model's performance. This trajectory predictor could accurately generate trajectories, outperforming the flight plan and other neural network approaches by a good margin. For demand prediction, introducing improved trajectories provided small gains that could lead to more stable predictions.**

***Keywords*—Trajectory prediction, transformer neural network, demand forecasting**

I. Introduction

Managing the balance between demand and capacity in air traffic is critical for ensuring safety and efficiency. Demand refers to the number of flights within a sector, while capacity denotes the Air Navigation Service Provider's (ANSP) ability to manage these flights safely. The Dutch ANSP, LVNL, utilizes flight plans to forecast demand, although the inherent uncertainty in flight data poses challenges to accurate forecasting. This research aims to refine air traffic sector demand forecasting through machine learning-based trajectory prediction, focusing on a three-hour lookahead time crucial for air traffic control decisions. By leveraging machine learning, the study seeks to provide a more accurate representation of flight trajectories, thereby improving predictability of air traffic demand within the Dutch airspace, with LVNL's support in data and expertise.

Effective demand forecasting is vital for balancing the number of aircraft in a sector against the available capacity. Traditionally, forecasts rely on flight schedules or plans, but deviations due to delays, re-routing, or disruptions introduce significant uncertainty. Könnemann [1] highlights departure time as a primary uncertainty source, with air traffic control (ATC) interference and trajectory prediction also contributing. Recent efforts aim to improve predictions through trajectory-based and aggregate approaches. Gilbo & Smith [2] achieved a reduction in demand prediction error for US sectors using regression models. Fernández et al.[3] utilized Hidden Markov Models for 4D trajectory predictions, demonstrating how these predictions can optimize demand and capacity by adjusting flight start times. Aggregate methods, treating sectors as traffic flow blocks, have shown predictive improvements, especially in short-term forecasts, with Ma et al.[4] employing a spatiotemporal graph network to enhance accuracy.

Accurate trajectory prediction underpins effective demand forecasting. Increased access to surveillance data and machine learning advancements have facilitated the shift towards data-driven methods. LSTM networks, for example, have shown promise in predicting accurate trajectories over significant lookahead times, as evidenced by the work of Overkamp [5], who demonstrated their effectiveness in free-route airspace. Clustering has also been employed to enhance trajectory prediction accuracy, with methods proposed by Fernández et al.[3] and Wu et al.[6] showing an improved performance by grouping similar flight trajectories before model application.

The introduction of transformer neural networks by Vaswani et al.[7] represents a significant advancement, offering parallel computation capabilities and efficiency improvements over traditional models. While initially designed for language processing tasks, transformers have shown potential in other domains, including traffic and pedestrian trajectory prediction by Wang et al.[8] and Achaji et al.[9]. Recent developments in token mixing algorithms, such as the Fast Fourier Transformation layer by Lee-Thorp et al.[10] and the multilayer perceptron by Tolstikhin et al.[11], offer promising alternatives to the attention mechanism, potentially reducing computational demands while maintaining high model performance.

This paper adopts the transformer neural network for the trajectory prediction task. We demonstrate a customized loss function, which is the key to the successful training of such neural networks for trajectory prediction. The structure of this paper is as follows: section II explains the chosen methodology and experiment set-up. In section III, the experiment results are provided, after which section IV is a discussion on the observed outcome. To end, section V contains the conclusion of this research paper.

II. Methodology

This section addresses the data processing, transformer model, customized loss function, and experiment set-ups for evaluating demand forecasts.

A. Input data

Since this research is relevant to operational decision-making, the selected input data must be readily available in the operational domain. For this reason, the following data sources were consulted to construct the input and testing datasets:

1) *Eurocontrol B2B flight messages*: The Eurocontrol B2B connection sends flight status messages to LVNL containing

information such as a flight plan, departure status, origin, destination, aircraft registration, and airline. In addition, it may also include timings, such as the estimated off-block time, taxi time, and estimated arrival time. This data is currently used to make demand forecasts for the coming 3-5 hours and has high integrity. Flight messages from May 2021 are available for this study.

2) *OpenSky ADS-B trajectories*: To train a model to predict trajectories, the actual flown trajectories must be specified. Because LVNL does not have surveillance data for complete origin-to-destination flights, this data must be sought elsewhere. ADS-B surveillance data is tracked and stored by several contributors on the collaborative OpenSky network. This data includes positions, speed, track, origin, destination, and aircraft identification.

3) *ERA5 meteorological data*: Various studies have shown the importance of including meteorological conditions in trajectory prediction models [12]. The ECMWF historical database provides meteorological parameters such as wind, temperature, and precipitation on 1000-1hPa pressure levels. The spatial resolution of this data is 31km and spans globally. For this research, the atmospheric parameters are taken at 0:00, 06:00, 12:00, and 18:00 UTC for the pressure levels from 1000hPa to 125hPa. The northerly and easterly wind components are taken, as well as the temperatures.

4) *Airspace data*: To make demand forecasts, trajectories must be overlaid with the relevant airspace block to test when and where the airspace is crossed. The airspace layout is taken from the Eurocontrol NM Demand data repository.

B. Data processing

Data is prepared and converted into a tensor format compatible with the PyTorch library. This involves resampling flights to a 4-minute temporal resolution, zero-padding to 220 samples, and integrating weather conditions into B2B flight messages. Next, these messages are aligned with ADS-B trajectories using flight numbers and departure times. Due to data volume and computational limits, the study focuses on a 3-hour lookahead time, selecting the last flight message before this period as the model input.

The process also entails formatting, detailing, and splitting the data into training and test datasets. The test set comprises 10 full traffic days, chosen randomly to evaluate both trajectory prediction and demand forecasting. Given the need to assess demand forecasting error across multiple flights, 30% of the dataset is allocated to testing, leaving the remainder for training. This allocation precludes a separate validation set, but training includes periodic testing to monitor for overfitting.

Formatting data correctly is crucial in machine learning, as models interpret variables solely through numerical values. Inputs are typically normalized to a 0-1 range to constrain the network effectively.

Normalizing parameters like flight plan and trajectory coordinates, however, is complex due to their representation in a spherical coordinate system, which poses challenges for neural networks due to its non-linear nature, as noted by Overkamp [5] and Tran et al.[13]. To address this, coordinates (latitude ϕ, longitude λ, and height h) are initially converted to the ECEF frame, then to the ENU frame centered on Amsterdam Airport (EHAM).

This process involves transformations detailed in Equation 1, Equation 2, and Equation 3, with the reference location being Amsterdam Airport. Subsequently, ENU coordinates are normalized to a 0-1 scale (with Amsterdam Airport as the upper bound) via a specific transformation matrix A, detailed in Equation 4. This normalization, which initially adjusts the range to 1-2 before shifting, is crucial because a direct 0-1 range lacks a unique inverse matrix. The transformation normalizes the ADS-B trajectory data for model input but also aids in converting model output back to real-world coordinates, potentially obviating the need for further dimensionality reduction techniques like clustering.

$$\begin{aligned} X_c &= N(\phi) + hcos\phi cos\lambda \\ Y_c &= N(\phi) + hcos\phi sin\lambda \\ Z_c &= N(\phi) + hsin\phi \end{aligned} \tag{1}$$

$$N(\phi) = \frac{a^2}{\sqrt{a^2cos^2\phi + b^2sin^2\phi}} \tag{2}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -sin\lambda_r & cos\lambda_r & 0 \\ -sin\phi_r cos\lambda_r & -sin\phi_r sin\lambda_r & cos\phi_r \\ cos\phi_r cos\lambda_r & cos\phi_r sin\lambda_r & sin\phi_r \end{bmatrix} \begin{bmatrix} X_c - X_r \\ Y_c - Y_r \\ Z_c - Z_r \end{bmatrix} \tag{3}$$

$$A = \begin{bmatrix} x_{origin} & y_{origin} \\ x_{EHAM} & y_{EHAM} \end{bmatrix}^{-1} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{4}$$

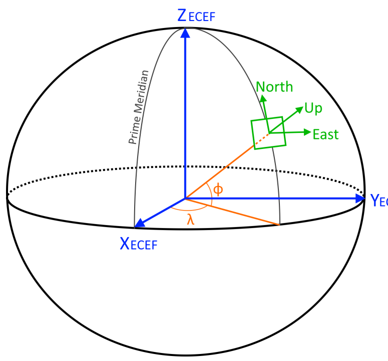


Figure 1: Reference frames used for geographic positions: Yellow is the spherical coordinate system. Blue is the ECEF reference frame. Green is the Cartesian ENU reference frame.

In addition to normalization, certain variables necessitate distinct formatting techniques. Static variables like origin airport, airline operator, and aircraft type are numerically encoded as integers, each representing a unique entity. Temporal variables, such as day of the week and departure time, undergo cyclical encoding to address the limitations of min-max normalization with cyclic data. For instance, integer encoding could misleadingly suggest a significant difference between Sunday and Monday, with values of 6 and 0, respectively. To accurately reflect cyclical relationships, Equation 5 is employed, generating two variables to prevent the ambiguity caused by sinusoidal encoding's symmetry. The complete list of features in the dataset are detailed in Tables I., II., and III..

$$H_{sin} = sin(\frac{2\pi H}{max(H)})$$
$$H_{cos} = cos(\frac{2\pi H}{max(H)}) \quad (5)$$

TABLE I.: Input Tensor Time Series Features

Time Series Feature	Processing	Size
FPL duration timestamp	-	220
FPL latitude	Trajectory normalization	220
FPL longitude	Trajectory normalization	220
FPL altitude	Min-Max normalization	220
North wind component	Min-Max normalization	220
East wind component	Min-Max normalization	220
Air temperature	Min-Max normalization	220

TABLE II.: Input Tensor Static Features

Static Feature	Processing	Size
Estimated take-off time	Cyclical Encoding	2
Day of the week	Cyclical Encoding	2
Pre-departure delay	Min-Max normalization	1
Origin airport	Integer encoding	1
Aircraft Type	Integer encoding	1
Aircraft Operator	Integer encoding	1

TABLE III.: Target Tensor Time Series Features

Time Series Feature	Processing	Size
ADS-B duration timestamp	-	220
Δ(ADS-B, FPL) latitude	Trajectory normalization	220
Δ(ADS-B, FPL) longitude	Trajectory normalization	220
Δ(ADS-B, FPL) altitude	Min-Max normalization	220

C. Trajectory prediction model

Based on the literature survey, the transformer neural network was selected as the most suitable candidate for the trajectory prediction task. The original transformer neural network was developed by Vaswani et al.[7] and consists of an encoder-decoder structure as shown in Figure 2.

The left side of the figure shows the encoder, and the right column shows the decoder. The original transformer had to be adapted slightly for the trajectory prediction task, which will be explained in more detail. Nonetheless, this was kept to a minimum in order to properly evaluate the potential performance of the transformer network. Looking over the elements of the transformer neural network in Figure 2, the encoder input is first processed by a linear input embedding layer. Secondly, a positional embedding is added so the model can relate the relative positions of input tokens. The positional encoding is sampled from a sinusoidal relation as shown in Equation 6.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \quad (6)$$

The following encoder layer consists of three sub-layers: A multi-head attention layer, an addition and normalization layer, and a feedforward layer. As the scheme shows, the input signal is duplicated to a bypass channel for each functional layer and

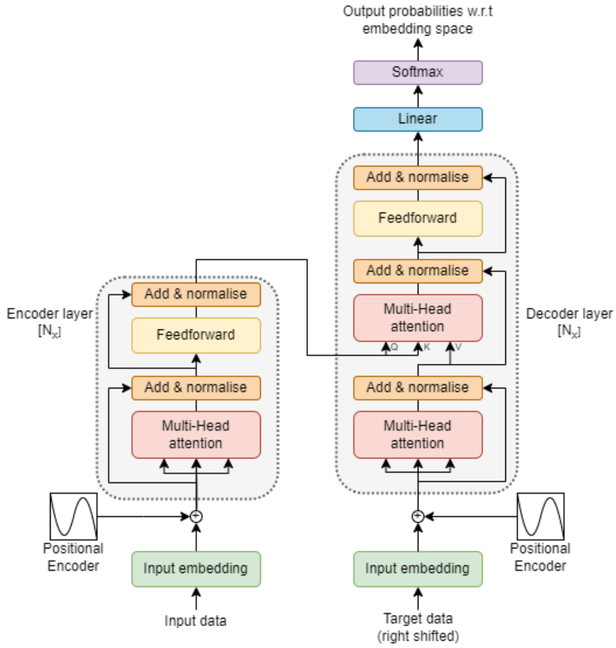


Figure 2: Schematic overview of the transformer neural network as proposed by Vaswani et al.[7].

added afterward, ensuring that the model does not suffer from vanishing or exploding gradient effects.

When looking at the decoder layer, various elements are similar to the encoder structure and need no further clarification. However, two primary differences are apparent. The signal that enters the decoder in the original transformer is the output signal shifted to the right and masked. This allows the model to complete the signal auto-regressively, similar to recurrent neural networks. Furthermore, the second multi-head attention layer in the decoder is different because the keys and queries are taken from the encoder output signal here.

In contrast, the values are obtained from the decoder's internal signal. After the multi-headed attention layer, a linear layer processes the decoder signal into a signal of the desired shape. The final soft-max element converts this signal to a probabilistic scale. In the original use case, the highest probability value determined which word was selected from the embedding space.

This operation is removed for trajectory prediction because the output can directly be translated into a trajectory via the transformation matrix used to normalize the input data. This, in fact, acts as the embedding space. The number of decoder layers in the original transformer is determined at four, but this hyper-parameter can be changed when iterating on the model architecture.

Most flights are not yet airborne because of the desired critical lookahead time of 3 hours for demand predictions. This poses the problem that flights must be predicted with actual trajectory data points available. It is, therefore, decided to build the trajectory prediction model primarily for a generative case, relying on flight plan input for both encoder and decoder. Nevertheless, an alternative model iteration is made where the transformer is trained for both generative and regressive use cases. In this auto-regressive model, the encoder input signal also receives the flight data points available.

Besides the classic transformer neural network, an adapted version is also created based on empirical testing and evaluation of different transformer neural network layouts. In order to compare the results of the transformer neural network, simple trajectory predictors are built based on various machine learning methods. The test cases include an LSTM network and a feedforward neural network. Each of these networks has four layers with 2048 neurons. This corresponds to the classic transformer neural network's layer complexity, allowing equal comparison.

D. Training with a customized loss function

After establishing the model layout and parameters, the training phase can begin. Neural networks have variable weights or parameters that each partly contribute to the outcome of the network. With a loss function, the error of the model can be calculated. Via error back-propagation, the specific gradients of each weight can then be determined.

In this research, a new customized loss function is designed. For the trajectory prediction problem, this custom loss function is specified based on requirements and iterative modeling experience. The loss function is given in Equation 7 and Equation 8.

$$\lambda = RMSE(\hat{y} - y) \tag{7}$$

$$\begin{aligned} L = &\lambda * W_1 + \lambda' * W_2 + \lambda_{alt} * W_3 + \lambda_{begin} * W_4 \\ &+ \lambda_{end} * W_5 + \lambda_{cruise} * W_6 \end{aligned} \tag{8}$$

where y and $\hat{y}$ are target and estimated variable. λ denotes the root mean square error for a given part of the output data or derivative. These weights were determined empirically to emphasize certain parts of the trajectory more during training. Table IV. provides the values of the weights in the loss function.

TABLE IV.: Loss function weights

Diff	Weight	Value	Description
λ	W_1	10	Entire trajectory error
λ'	W_2	10	Derivative trajectory error
λ_{alt}	W_3	10	Altitude error
λ_{begin}	W_4	3	First three datapoints error
λ_{end}	W_5	5	Final 20% of trajectory error
λ_{cruise}	W_6	2	50-70% of trajectory error

Unlike common trajectory prediction models, the target variable y is set to be the difference between the filed flight plan points and the actual trajectory. It is found that this gives more stable trajectory prediction results than directly predicting the actual trajectory. As a result, the final output vector of the model ($\hat{y}$) must be added to the input signal to get the actual predicted trajectories. W denotes the weights assigned to the specific error contribution in the loss function.

E. Experimental set-up for demand forecasting use case

The trajectory prediction and the demand forecasting performance are tested to evaluate the suitability of the proposed transformer model.

The validation dataset consists of 10 full days of traffic randomly selected from the month of available data. For the trajectory prediction, 2839 flights can be used to assess the TP performance. During training, the loss function measures the predictive accuracy (Equation 8).

In the experiment, lateral predictive accuracy is measured in along-track, cross-track, and horizontal error, as shown in Figure 3. Only the flight's airborne part is modeled, meaning that all trajectories take off at t_0. The errors are then calculated at each timestamp, comparing the actual trajectory to the predicted trajectory. Similarly, the altitude difference at each timestamp is evaluated.

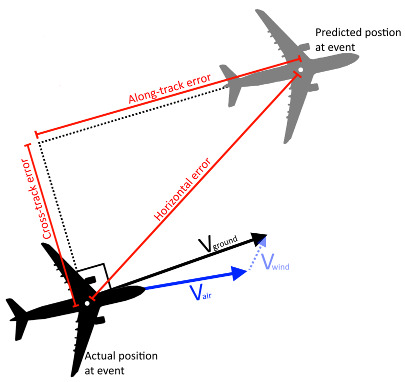


Figure 3: Horizontal error metrics for trajectory prediction.

The mean absolute error for each metric is taken over the entire trajectory. Another important metric to compare is the total trajectory distance, which shows whether the predicted trajectory is globally coherent. In the experiment, the error distribution of the different models will be compared to the baseline trajectory, which is the flight plan.

The demand prediction experiment considers the predicted trajectories to determine the expected demand in the Amsterdam FIR, which is first calculated. Arrival time is one of the driving variables; hence, this is an important metric to consider.

The evaluation of arrival time will be two-folded: Since the predicted trajectories are trained based on aligned take-off times, there is no pre-departure delay predicted by the TP model. This was a considerate design choice because ground-based delay has a significantly different complexity than airborne delay. This was expected to introduce much uncertainty into the model had it been included while requiring an architectural change for both the model and data format. Because of this, the trajectory predictor model generates a trajectory that starts at the given departure time in the input dataset.

Therefore, the predictions and actual flights are compared based on flight duration up to the FIR arrival time. This metric is most important to determine the impact of the airborne element on demand forecast. However, when predicting demand, absolute predicted arrival times are the input; hence, the absolute arrival time prediction accuracy is also evaluated in the experiment.

Ultimately, demand forecasts are the most critical metric for tactical decision-making. Hence, these are pivotal to evaluate in the experiment as well. Currently, the demand window is 20 minutes, renewed every 5 minutes. As a result, this experiment considers the same window size and refresh rate. The demand predictions for the 10 days in the test dataset are compared to the actual demand.

It is to be noted that this research is scoped to predictions for the situation 3 hours ahead only and that some flights with limited input or actual trajectory data are filtered out. Hence, the results are not an operational demand forecast but only an assessment of the 3-hour lookahead time. However, since this research is constrained to pre-departure generative trajectory predictions, a direct comparative analysis of the demand forecasting performance can still be made.

III. Results

After training several model varieties sufficiently, the model performance can be assessed. This section presents the results of the developed transformer neural network compared to other methods. In the first subsection, the trajectory prediction performance is assessed. The second subsection shows the model's performance when forecasting air traffic demand.

A. *Trajectory prediction performance*

Before showing the predictive capabilities, a summary of the model training efforts is given in Table V.. With the loss values that were found, it can be expected that the improved transformer seems to be the best-performing model. Moreover, this model trains relatively fast and is stable up to 2500 epochs.

TABLE V.: Model training results

	TF AR	TF improved	TF classic	LSTM	MLP
Trainable parameters	24E6	24E6	30E6	117E6	17E6
Time per epoch [s]	33	32	37	39	16
#epochs trained	2500	2500	1600	500	200
Final loss	35	30	39	48	65
Overfitted	No	Yes	No	Yes	Yes

Although training results are important to assess model performance, the experiment on the test dataset will show whether or not the model can improve trajectory predictions. During the first analysis of the experiment, it was found that none of the models produced accurate trajectory predictions for long-haul flights. This limitation was especially relevant for flights longer than 6 hours, which make up approximately 30% of the dataset. Therefore, it is decided to include only flights with an estimated flight time below this threshold. This implications are further discussed in section IV. The final results after having applied this filter are given below. In Figure 6, the mean absolute along-track, cross-track, and horizontal error distribution are shown. Figure 4 presents the mean altitude error. Finally, Figure 5 shows the total trajectory distance error.

When comparing the different models to the predictive performance of the flight plan, it can be noted that the feedforward model has significant difficulty in predicting trajectories. This is expected because feedforward neural networks are only sometimes applied to sequential data structures. The LSTM neural network shows some improvements compared to the flight plan, but it was found to be sensitive to overfitting during training. After 500 epochs, the training was stopped because

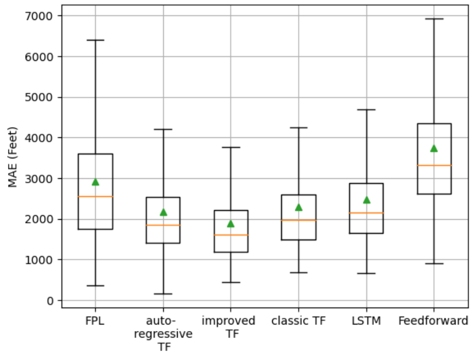


Figure 4: Mean absolute altitude error of the different TP models.

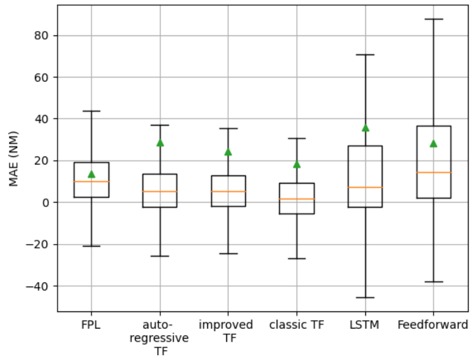
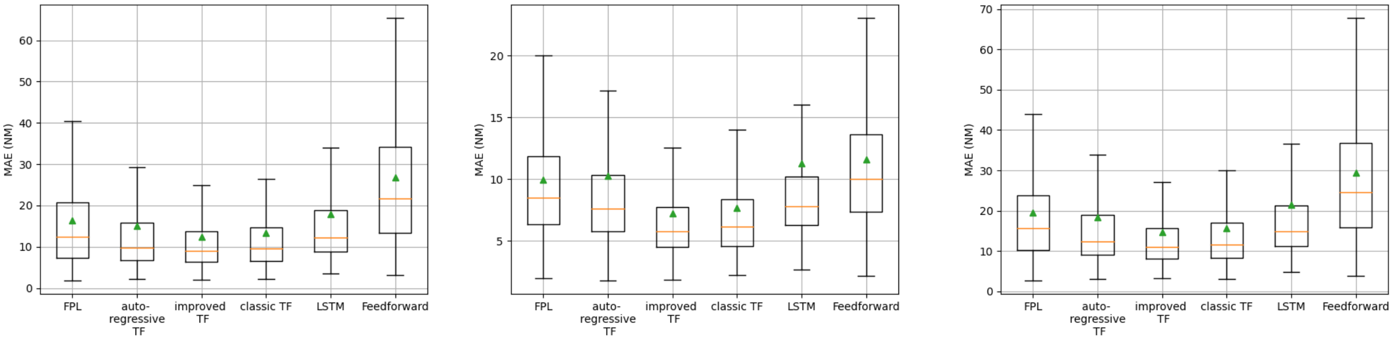


Figure 5: Mean total trajectory distance error of the different TP models.

validation results became worse. The classic transformer neural network was not sensitive to overfitting but converged more slowly. The final model was trained to 1600 epochs until the loss stabilized. However, no signs of overfitting were found during intermittent validation. The transformer outperforms the three baseline models with a lower mean error and a reduced error distribution. Only for the global distance attribute, all machine learning models fail to reduce the mean error, hinting at several trajectories with very large errors. This is not uncommon for a data-driven model and does not affect the majority of predictions.

Based on literary findings and the obtained results from the three baseline models, the semi-optimized transformer neural network was created through iterative training of various model architectures. In this network, the decoder complexity was reduced to a single layer only, as it was found that the level of complexity in the data was not easily captured through the auto-encoder structure. However, a linear pipeline performed better and could be trained further. Moreover, the encoder complexity was increased to 6 layers, and the final output block was extended with a linear and ReLu activation layer.

The results of this model are a significant improvement from the baseline models. Predictive accuracy horizontally and vertically improved to a lower median error and better distribution. This shows that the transformer neural network can explain the differences between the filed flight plan and actual flight execution. Looking at the altitude predictions, when observing individual trajectories, most of the errors are reduced by more accurately predicting climb and descending phases. The cruise phase prediction is reasonable, but the flight



(a) Along-track error distribution (b) Cross-track error distribution (c) Horizontal error distribution

Figure 6: Distribution of the lateral trajectory prediction error, where the mean absolute error is taken for every flight.

plan is usually found to be more accurate here. An example of a predicted trajectory is shown in Figure 7. Finally, the improved transformer was adapted to an autoregressive variant capable of extrapolating future data points of an already airborne trajectory. Although it still improves TP accuracy, the model failed to surpass the accuracy of the fully generative models. However, the presented trajectory prediction improvements are significant overall, especially for the improved generative transformer neural network.

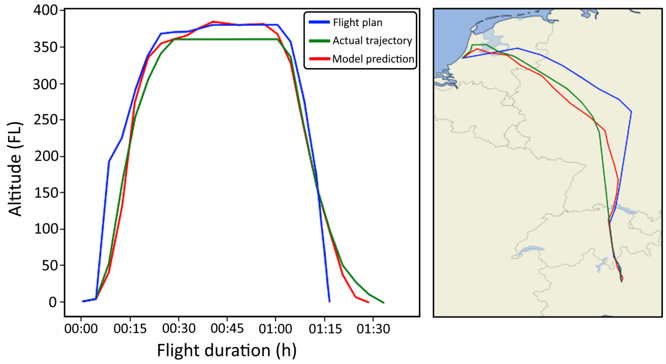


Figure 7: Example of a prediction for a flight between Milan and Amsterdam. The predicted trajectory is compared to the field flight plan and the actual trajectory.

B. Demand forecasting performance

With the observed increase in trajectory prediction accuracy, demand forecasts can benefit from more accurate predictions of aircraft entering the target airspace. Demand forecasting performance is measured with a variety of metrics, amongst which the flight duration time up to FIR entry, arrival time, and demand. The validation dataset includes 10 days of predicted and actual trajectories used to evaluate the performance. First, because only airborne segments of flights are considered, the predicted flight duration times up to FIR entry must be evaluated. This shows whether or not the TP model is likely to make an improved arrival time estimate and, henceforth, is suitable for demand forecasting. These results are shown in Figure 8. As expected, all models that showed improved TP accuracy also estimated the flight duration better than the flight plan. Secondly, the absolute arrival times at the FIR boundary are calculated with the flight's duration and the expected take-off time from the B2B message. These absolute arrival times are then compared to the actual arrival times from the ADS-B trajectories. The absolute arrival time prediction errors are presented in Figure 9.

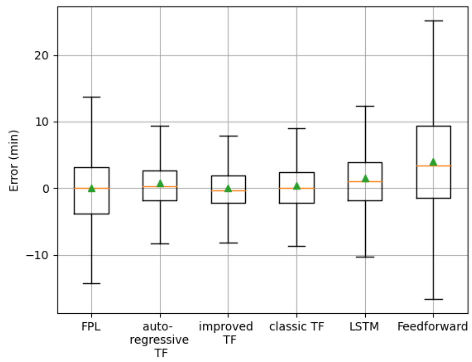


Figure 8: Error between actual flight duration and the predicted flight duration up until FIR entry.

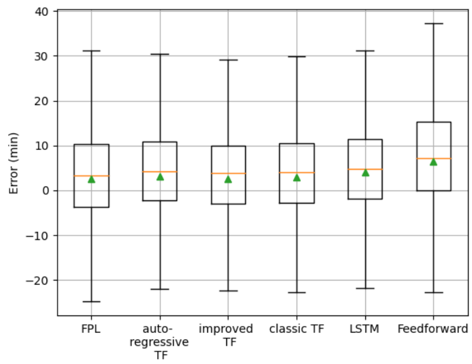


Figure 9: Error between actual FIR and predicted arrival times.

The various models' predicted arrival times of flights do not show the same increase in accuracy as observed in the flight duration evaluation. This is partly expected because the take-off time uncertainty is known to be a considerable factor, which was argued by Könnemann [1] amongst others. However, some improvements can still be observed, as the spread of errors is reduced to some extent with the transformer models. Finally, the demand forecasts of each method are evaluated. LVNL considers the demand forecast 3 to 5 hours ahead to be the most valuable in current operations. This allows the ANSP to apply the required capacity resources or to regulate incoming traffic. For this reason, the dataset is constrained to B2B messages 3 hours before the actual arrival of the flight. The demand forecast results of all different trajectory predictor models are presented in Table VI..

TABLE VI.: Demand forecast model results. Measured by the difference in the number of flights predicted to enter the airspace and the actual number of flights.

	RMSE	MAE	R2	std
FPL	1.83	1.12	0.79	1.83
Autoregressive TF	1.73	1.07	0.81	1.73
Improved TF	1.66	1.04	0.82	1.66
Classic TF	1.78	1.10	0.80	1.78
LSTM	1.83	1.13	0.78	1.83
Feedforward	2.1	1.32	0.72	2.09

Comparing the demand errors from the flight plan-based method to the tested TP models, it becomes clear that most models do not provide any significant improvements. The feedforward model performs poorly, which was expected based on the trajectory prediction accuracy. The LSTM model had slightly better TP accuracy than the flight plan, but these improvements do not lead to any benefits in the demand forecasting case.

The transformer neural networks show slightly improved demand forecasts. The improved transformer shows a more significant jump over the flight plan-based method. The improved transformer model has a root mean square error that is 10% lower than the flight plan-based approach. Compared to the results of the other models, this is the only significant improvement, given the dataset size of 10 days. When looking at the demand error distribution in Figure 10, it becomes clear that the improved transformer marginally reduces the peak errors. Hence, the improved trajectories potentially provide a more stable demand forecast.

Nevertheless, the differences are very small. To further clarify the results, Figure 11 shows the demand forecast for an entire day of traffic. The traffic peak predictions seem less erroneous compared to the flight plan-based approach, which is a desirable improvement.

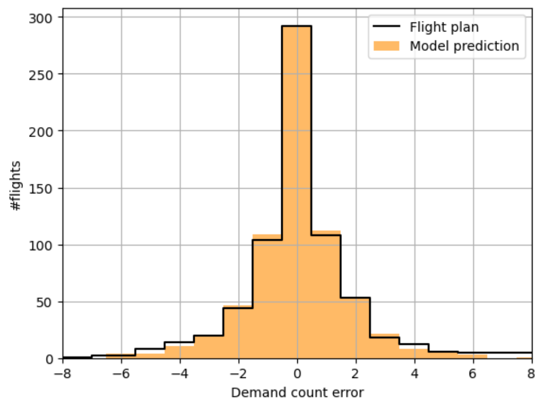


Figure 10: Distribution of predicted demand error for the flight plan and the improved transformer approach.

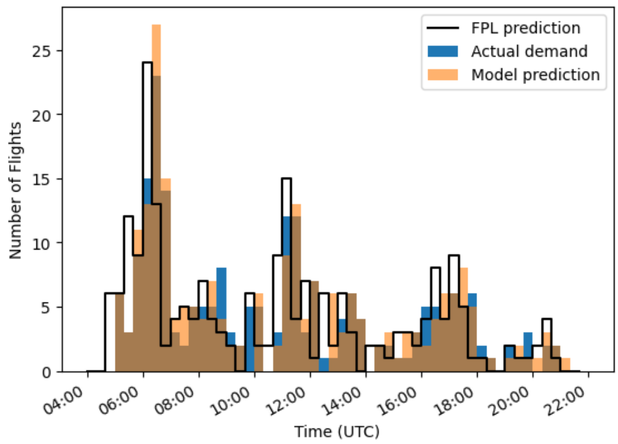


Figure 11: Example of predicted demand on May 19, 2021, where each bin is predicted with a 3-hour lookahead time.

IV. Discussion

A. *Generative machine learning model for trajectory prediction*

In evaluating the suitability of generative machine learning methodologies for trajectory prediction, it was observed that while generative models struggled with flights over 6 hours due to normalization issues, they excelled in predicting trajectories for flights under this duration. The generative data-driven approach, particularly with the transformer and improved transformer models, showed significant promise in enhancing long-term trajectory prediction accuracy, outperforming traditional LSTM networks and feedforward neural networks, despite the latter's inability to handle sequential data effectively.

The auto-regressive version did not meet expectations, possibly due to limited contextual data from shorter flights. Comparatively, the transformer models achieved superior precision, with the improved transformer notably reducing mean absolute errors significantly more than previous generative LSTM models, as evidenced by Liu and Hansen's findings [12].

The success of these models from this paper is attributed to effective data formatting, normalization, and a custom loss function, which simplified data variance and improved training outcomes. Additionally, the transformer's multi-headed attention mechanism and the omission of a complex encoder in the improved version facilitated more stable training and promising results, suggesting potential areas for further optimization in future iterations.

B. *Suitability of prediction for air traffic demand prediction*

The improved trajectory predictions facilitated more precise demand forecasts for the 3-hour lookahead time, enhancing the accuracy of flight duration estimates significantly, with the improved transformer model reducing the interquartile range by nearly 30% compared to flight plan-based approaches. However, improvements in predicting arrival times were less pronounced, possibly due to the models' lower performance on flights with significant deviations from planned operations, influenced by factors like pilots' responses to delays.

Despite receiving updated departure time and delay information, the models might not fully capture the impact of such operational variances. The complexity of accurately predicting non-standard flight behavior suggests that the neural networks used might either be inadequate for identifying correlations between flight path changes and ground delays or the training data was insufficiently representative of such flights.

Regarding overall demand forecast accuracy, most models showed no significant improvement, except for the improved transformer neural network, which notably enhanced prediction accuracy, particularly in predicting peak demand periods. This improvement is critical for air traffic management, as accurate peak demand forecasts contribute to more efficient flow management. However, to validate these findings, a larger test dataset is necessary for a comprehensive analysis.

C. Implications and limitations

The chosen generative trajectory prediction approach, particularly with the semi-optimized transformer neural network, demonstrates notable enhancements in accuracy, albeit with a significant limitation for flights exceeding 6 hours due to decreased spatial resolution from normalization. A potential solution involves developing a dedicated model for long-haul flights. For demand forecasting, the predictions yield improved stability, yet improvements are constrained without accurate departure time data.

Despite LVNL's success in reducing mean absolute error using a random forest regressor based on flight schedule data [14], this study's trajectory prediction (TP) method did not achieve similar enhancements in demand forecasting error reduction. Nonetheless, the error distribution for demand forecasts was narrowed, highlighting the value of including actual take-off times for airborne flights to potentially boost demand prediction accuracy, albeit at the cost of obscuring the direct impact of TP improvements.

V. Conclusion

This research aimed to enhance air traffic sector demand forecasting by employing machine learning techniques for trajectory prediction, focusing on the transformer neural network. This study is part of a broader initiative to ensure more sustainable, safe, and efficient airspace navigation as the global Air Traffic Management (ATM) system evolves towards comprehensive flight trajectory management. ANSPs, including LVNL, aim to align air traffic demand with available airspace capacity. This task is challenging due to the unreliability of flight plans and schedules for accurate demand forecasting hours ahead.

Data for this study was sourced from Eurocontrol NM B2B feeds and the OpenSky ADS-B repository, encompassing filed flight plans, flight status information, and actual flight trajectories. This data, prepared and normalized to facilitate a direct comparison of trajectory prediction performance, underpins the development of both trajectory prediction (TP) and demand forecasting models. Various models, including a conventional transformer neural network, LSTM, and feedforward neural network, were evaluated, creating an improved transformer model and an auto-regressive version for enhanced predictive accuracy.

The analysis revealed that transformer neural networks, particularly the semi-optimized version, significantly improved trajectory prediction and demand forecasting accuracy for flights up to 6 hours long. However, the normalization method may have limited predictions for longer flights, suggesting room for model-specific improvements. Despite challenges in accurately predicting actual arrival times, advancements in trajectory prediction resulted in more accurate flight time estimates and demand forecasts, indicating that further refinement in departure time estimates could augment this trajectory-based forecasting approach.

Acknowledgment

We would like to thank Ferdinand Dijkstra from the Knowledge and Development Centre for the support of research.

References

[1] E. Könnemann, “Performance impact of improved departure time prediction relative to sector demand & arrival time predictability,” *TU Delft Repository*, December 2014.

[2] E. Gilbo and S. Smith, “A new model to improve aggregate air traffic demand predictions,” *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2007.

[3] E. C. Fernández, J. M. Cordero, G. Vouros, N. Pelekis, T. Kravaris, H. Georgiou, G. Fuchs, N. Andrienko, G. Andrienko, E. Casado, *et al.*, “Dart: A machine-learning approach to trajectory prediction and demand-capacity balancing,” *Seventh SESAR Innovation Days, Belgrade*, November 2017.

[4] C. Ma, S. Alam, Q. Cai, and D. Delahaye, “Sector entry flow prediction based on graph convolutional networks,” in *International Conference on Research in Air Transportation*, 2022.

[5] J.-L. Overkamp, “Long short-term memory network based trajectory prediction incorporating air traffic dynamics,” *TU Delft Repository*, September 2021.

[6] Y. Wu, H. Yu, J. Du, B. Liu, and W. Yu, “An aircraft trajectory prediction method based on trajectory clustering and a spatiotemporal feature network,” *Electronics*, vol. 11, p. 3453, October 2022.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., December 2017.

[8] Z. Wang, J. Guo, Z. Hu, H. Zhang, J. Zhang, and J. Pu, “Lane transformer: A high-efficiency trajectory prediction model,” *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 2–13, January 2023.

[9] L. Achaji, T. Barry, T. Fouqueray, J. Moreau, F. Aioun, and F. Charpillet, “Pretr: Spatio-temporal non-autoregressive trajectory prediction transformer,” March 2022.

[10] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontañón, “Fnet: Mixing tokens with fourier transforms,” *CoRR*, vol. abs/2105.03824, May 2022.

[11] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, “Mlp-mixer: An all-mlp architecture for vision,” May 2021.

[12] Y. Liu and M. Hansen, “Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach,” December 2018.

[13] P. N. Tran, H. Q. V. Nguyen, D.-T. Pham, and S. Alam, “Aircraft trajectory prediction with enriched intent using encoder-decoder architecture,” *IEEE Access*, vol. 10, pp. 17881–17896, February 2022.

[14] L. Nederland, *Software Design Description (SDD), Decision Support Tool - Load Predictor*, 2.0 ed., 2020.