



HAL
open science

An Improved ML Method to Speed Up the Trajectory Prediction: Taking Melbourne Airport as a Study Case

Yuting Xi, Ji Ma, Zhengyi Wang, Hongyang Zhang, Alessandro Gardi,
Roberto Sabatini, Daniel Delahaye

► **To cite this version:**

Yuting Xi, Ji Ma, Zhengyi Wang, Hongyang Zhang, Alessandro Gardi, et al.. An Improved ML Method to Speed Up the Trajectory Prediction: Taking Melbourne Airport as a Study Case. Asia-Pacific International Symposium on Aerospace Technology (APISAT 2023), 2, SPRINGER, pp.1689-1699, 2024, 978-981-97-4012-3. 10.1007/978-981-97-4010-9 . hal-04665660

HAL Id: hal-04665660

<https://enac.hal.science/hal-04665660v1>

Submitted on 31 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Improved ML Method to Speed Up the Trajectory Prediction: Taking Melbourne Airport as a Study Case

Yuting Xi¹, Ji Ma², Zhengyi Wang³, Hongyan Zhang⁴, Man Liang^{1*}, Alessandro Gardi^{5,1}, Roberto Sabatini^{5,1}, Daniel Delahaye⁶

¹RMIT University, Melbourne VIC 3000, Australia

²CAUC-ENAC Joint Research Center of Applied Mathematics for Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China

³Air Traffic Management Research Institute, Nanyang Technological University, Singapore

⁴School of Information Science and Technology, Hainan Normal University, Haikou, China

⁵School of Aerospace Engineering, Khalifa University, Abu Dhabi, UAE

⁶ENAC, Université de Toulouse, Toulouse 31055, France

Abstract. The safety and efficiency of airspace operations largely depend on the accurate prediction of 4D trajectories in dense air traffic. Traditional methods are progressively giving way to more accurate machine learning (ML) techniques, among which the Long Short-Term Memory (LSTM) neural network emerges as an exceptionally promising tool and has been successfully applied, especially for time-series prediction tasks. In this study, we introduce an LSTM-based adjustable interpolation algorithm designed to significantly reduce computational time while maintaining accuracy at an acceptable level to meet operational constraints. The algorithm applies adjustable time intervals to input data based on ascent and descent rates, providing different data densities for different flight phases. A case study focusing on flight trajectories from Melbourne to Sydney is conducted, and the findings reveal that our proposed method can reduce computation time by half without significantly sacrificing prediction accuracy compared to the traditional linear interpolation method. Furthermore, it achieves accuracy improvements of at least 50% compared to raw data processing, with no substantial increase in computational time. Proven to be effective, our proposed algorithm can be an ideal solution for training dense air traffic data when regular training is required to meet accuracy and safety requirements. This includes applications in Urban Air Mobility (UAM) and unmanned aircraft operations, as well as airport management and airspace sector handovers.

Keywords: Air Traffic Management; Urban Air Mobility; Trajectory Prediction; Machine Learning; LSTM Neural Network, Adjustable Linear Interpolation Algorithm

1 Introduction

Urban Air Mobility (UAM) necessitates an on-demand operation capable of safely executing missions in metropolitan areas, encompassing emergency medical evacuations, rescue operations, humanitarian endeavors, and more [1]. Due to the stringent

safety requirements and the underdeveloped UAM infrastructure, current UAM operations rely extensively on pre-existing networks. To advance this concept and its operational efficiency further, it is imperative to devise novel technologies and algorithms based on the existing infrastructure and data to enhance safety and effectiveness [2]. One of the most daunting challenges in UAM pertains to the unpredictable behavior of Unmanned Aircraft Vehicles (UAVs), which can potentially pose environmental hazards and public safety concerns [3]. However, performance-based operation requires to establish a predictable and hazard-free aerial environment, in which, trajectory prediction plays a pivotal role, offering crucial support for conflict detection, maintain situational awareness and therefore reduce collision risk [4, 5]. Mathematical and Machine Learning (ML) models have both been considered for trajectory prediction. Mathematical models are better suited for real-time short-term trajectory prediction due to their algorithmic complexity [6] In contrast, ML, particularly Long Short-Term Memory (LSTM), exhibits significant promise in terms of prediction accuracy, thanks to its capacity to uncover hidden patterns in trajectory data [7]. However, several challenges persist, particularly concerning the lack of accurate data for ML-based trajectory prediction in the context of UAM. Considering the future of UAM operations, the widespread utilization of Automatic Dependent Surveillance-Broadcast (ADS-B) is anticipated. Vertical Take-off and Landing Aircraft (VTOL) flight profiles closely resemble those of commercial aircraft, which makes commercial aircraft ADS-B data a suitable case study to validate the proposed methodology for application in UAM. Furthermore, adequately training an ML model demands a substantial training dataset, which often results in extended computation times, a pressing concern for UAM [8]. To expedite the training process while maintaining prediction functionality and accuracy at safe levels, this paper introduces a trajectory prediction algorithm founded on LSTM and linear interpolation. This algorithm enables real-time trajectory prediction by dynamically adjusting data density based on data significance and optimizing training efficiency. The case study demonstrates that the algorithm achieves a balance between training time and accuracy, offering the potential to serve as a foundational element for future UAM trajectory prediction systems.

2 Literature Review

Due to the increasing density of current airspace, there is an urgent need for trajectory prediction to assist air traffic management. Trajectory prediction in air mobility typically involves tracking UAVs' current and previous positions to forecast their future movements [9]. Compared to ground traffic prediction, predicting air trajectories is more challenging due to the significant influence of unpredictable weather and complex airspace constraints [10]. Multiple trajectory prediction algorithms have been developed, utilizing both Machine Learning (ML) and mathematical approaches. Given the complexity of mathematical methods, ML has gained significant attention due to its ability to identify and adapt to patterns in data. Among ML algorithms, Deep Neural Networks (DNNs) are particularly powerful [11, 12]. Its inherent structure,

featuring hidden layers and numerous neurons, has led to their successful application in various fields, including language modeling, machine translation, and speech recognition [13]. However, RNNs have their limitations. They suffer from a lack of long-term memory, making them less effective at learning from extensive historical data [14]. Additionally, training RNNs can be complex, and their accuracy is not always guaranteed [12]. This is where Long Short-Term Memory (LSTM), a variant of the RNN architecture, comes into play. LSTM is built around a cell state (C_t) that flows through all LSTM cells. LSTM can maintain and convey data that needs to be retained from the initial LSTM unit to the final one [14].

3 System Design

The objective is to develop a real-time model capable of accurately predicting sequential trajectory characteristics using LSTM based on trajectory information. Initially, data is sourced from commercial airspace due to the unavailability of urban airspace data. The proposed system design, as illustrated in Fig. 1, outlines the workflow. The first step involves the utilization of the proposed algorithm, Adjustable Linear Interpolation (ALI), as part of the pre-processing phase. ALI is applied to generate a more structured dataset. Subsequently, the data is processed using a sliding window technique to reshape it into training data.

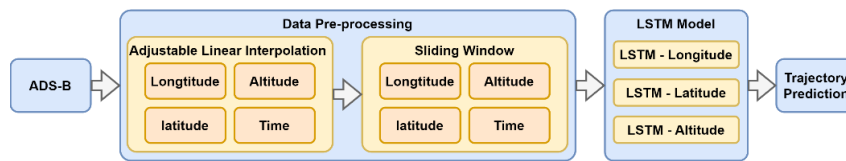


Fig. 1. System design.

3.1 Data Pre-processing

ADS-B data sourced from FlightRadar24 exhibits non-deterministic behavior, with time intervals fluctuating between 2s to 120s. This variability can significantly impact the accuracy of the model, leading to confusion in the LSTM model. Moreover, data redundancy can result in inefficient training, prolonging the model training time. Conventional Linear Interpolation (LI) addresses the issue of time interval fluctuation by converting it into a constant value, but it doesn't resolve the problem of data redundancy. To strike a balance between these concerns, the proposed system employs a novel linear interpolation method in which the time interval is determined based on the aircraft's ascent and descent rate. This approach effectively retains critical trajectory data while filtering out redundant information. The pseudocode below illustrates the Adjustable Linear Interpolation (ALI) algorithm, where T_1 , T_2 , T_3 , and T_4 represent different time intervals corresponding to the ascent/descent rate (R). Typically, these values from T_1 to T_4 should increase sequentially to accommodate varying air risks.

Algorithm 1 Adjustable Linear Interpolation

Input: \mathcal{T} : Set of time data, \mathcal{L}_A : Set of latitude data, \mathcal{L}_O : Set of longitude data, \mathcal{A} : Set of altitude data
Output: \mathcal{T}_N : Set of new time data, \mathcal{L}_{NA} : Set of new latitude data, \mathcal{L}_{NO} : Set of new longitude data, \mathcal{A}_N : Set of new altitude data

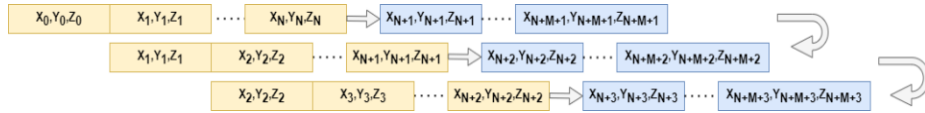
```

for n in range(0,  $\mathcal{T}$ ) do
   $R \leftarrow (\mathcal{A}_{n+1} - \mathcal{A}_n) / (\mathcal{T}_{n+1} - \mathcal{T}_n)$ 
  if  $R < 10$  then
     $TI \leftarrow T1$ 
  else if  $R \geq 10$  and  $R < 30$  then
     $TI \leftarrow T2$ 
  else if  $R \geq 30$  and  $R < 70$  then
     $TI \leftarrow T3$ 
  else
     $TI \leftarrow T4$ 
  end if
  Generate NTS from TI
end for
for t in NTS do
  for n in range(0,  $\mathcal{T}$ ) do
     $\mathcal{T}_N \leftarrow \mathcal{A}_n + \frac{\mathcal{A}_{n+1} - \mathcal{A}_n}{(\mathcal{T}_{n+1} - \mathcal{T}_n)(t - \mathcal{T}_n)}$ 
     $\mathcal{L}_{NO} \leftarrow \mathcal{L}_{O,n} + \frac{\mathcal{L}_{O,n+1} - \mathcal{L}_{O,n}}{(\mathcal{T}_{n+1} - \mathcal{T}_n)(t - \mathcal{T}_n)}$ 
     $\mathcal{L}_{NA} \leftarrow \mathcal{L}_{A,n} + \frac{\mathcal{L}_{A,n+1} - \mathcal{L}_{A,n}}{(\mathcal{T}_{n+1} - \mathcal{T}_n)(t - \mathcal{T}_n)}$ 
  end for
end for

```

3.2 Sliding Window Method

The sliding window is a crucial component of the data pre-processing phase, as depicted in Fig. 2. Its purpose is to collect the most recent n sequential trajectory data points and the subsequent m sequential trajectory data points at a specific timestamp t . As the sliding window traverses the entire dataset, trajectory data is divided into multiple sub-datasets, each containing a total of $m+n$ data points. Within each sub-dataset, the first n data points are designated as DataX, while the remaining data points are referred to as DataY. For instance, in the figure provided, the labels X, Y, and Z correspond to latitude, longitude, and altitude, respectively. The yellow dataset represents DataX, while the blue dataset represents DataY. The reshaped data is then fed into the LSTM model. Notably, the three input data dimensions—longitude, latitude, and altitude—are trained separately. To implement the system design outlined above, a dataset comprising 100 trajectories originating from Melbourne Airport to Sydney Airport was sourced from FlightRadar24. Of this dataset, 70% was allocated for training, while the remaining 30% was utilized for validation.

**Fig. 2.** Sliding window.

3.3 System Set Up

The LSTM models utilized for training altitude, longitude, and latitude are entirely identical. Each model is composed of two LSTM layers. The input layer is configured with 64 neurons, followed by a dense layer with 32 neurons, and finally, an output layer with 5 outputs. This LSTM model configuration enables the algorithm to predict the next five waypoints based on the previous twenty waypoints. To assess the accuracy of the training results, several error metrics are employed, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The average values of MAE, MAPE, and RMSE are calculated to determine the final error of the model. In the subsequent equations, K represents the number of rounds of training or testing conducted for a single trajectory. In this study, N is set to 5 since the model's objective is to predict 5 trajectory data points. P_{ji} denotes the i^{th} predicted trajectory data in round j , while A_{ji} signifies the i^{th} actual trajectory data in round j .

$$RMSE_{Av} = \frac{1}{K} \sum_{j=1}^K \sqrt{\frac{\sum_{i=1}^N (P_{ji} - A_{ji})^2}{N}} \quad (1)$$

$$MAE_{Av} = \frac{1}{K} \sum_{j=1}^K \frac{\sum_{i=1}^N |P_{ji} - A_{ji}|}{N} \quad (2)$$

$$MAPE_{Av} = \frac{1}{K} \sum_{j=1}^K \left(\frac{1}{N} \sum_{i=1}^N \left| \frac{P_{ji} - A_{ji}}{A_{ji}} \right| \right) \quad (3)$$

To determine the most suitable combination of T1, T2, T3, and T4 for the Adjustable Linear Interpolation (ALI) method, this paper investigated five different combinations and assessed their effectiveness and feasibility. The five combinations for ALI are presented below.

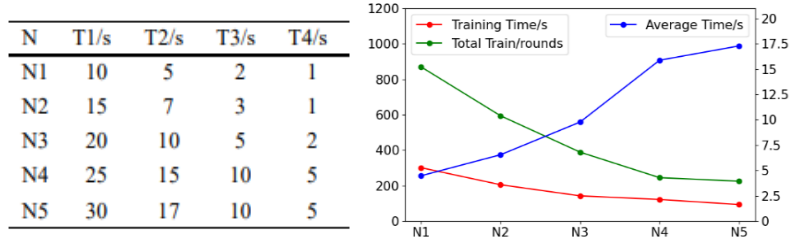


Fig. 3. Training efficiency for the five combinations.

In these combinations, N represents the code name for different combinations. Each combination comprises the four intervals T1-T4 as outlined in the pseudocode. Given the vast number of potential combinations, it is impractical to test them all. Therefore, the authors have selected the most reasonable ones. To ensure safety, the maximum time interval during the ascent/descent phase is capped at 5 seconds. Three factors are considered: training time, total training, and average time interval. In Fig. 3, "total train" indicates the number of training sessions conducted for each N , train-

ing time reflects the time required for each training session, and "average time interval" calculates the mean time interval for each combination. The objective is to identify an N that strikes a balance between low training time and a reasonable average time interval. The results clearly indicate that when training time and total training is low, the average time interval tends to increase significantly, which negatively impacts accuracy, and vice versa. Therefore, "N3" emerges as the best choice, striking a balance between these three factors, and it has been selected for implementation in this study.

4 Results

4.1 Training and Computation Time

In the training phase, longitude, altitude, and latitude are simultaneously trained using both pre-processing methods: ALI, LI, and raw data (RD). The ALI time interval is set as previously discussed, while the LI time interval is fixed at 5 seconds. The sliding window is configured with $n = 20$ and $m = 5$. Prior to the main training process, five trajectories were selected and pre-trained to simplify the training process and evaluate the impact of the ALI algorithm during training. Fig. 4 illustrates the training times for the three different pre-processing methods. It is evident that ALI reduces training time by almost 50% compared to LI. However, in comparison to RD, it results in an average increase of training time by approximately 30%. This experiment demonstrates that, when compared to LI, the ALI algorithm proves to be beneficial for future trajectory prediction development. To further assess the accuracy of each method, models equipped with either ALI, LI, or RD are separately trained using 70 trajectories.

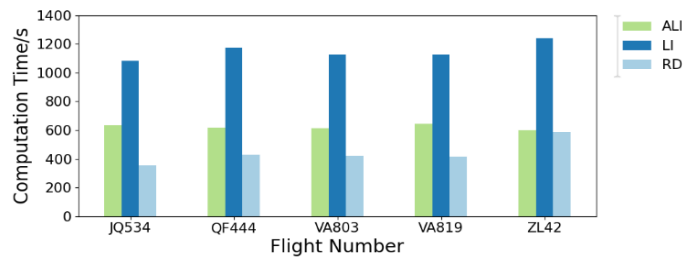


Fig. 4. Training efficiency.

4.2 Test Results

To gain a more intuitive understanding of the errors associated with the three methods, a random trajectory was selected for verification purposes. As depicted in Fig. 5, the predictions are plotted concurrently with the actual trajectory for visualization. In the plots, the green line represents the actual trajectory that requires prediction, and the points along the green line represent the predictions generated by the model. The

X and Y axes correspond to latitude and longitude, measured in degrees, while the Z axis represents altitude in feet. From a visual standpoint, LI and ALI exhibit similar levels of accuracy, although LI predictions tend to be more precise. ALI tends to experience more errors when transitioning between phases, and RD exhibits the lowest accuracy.

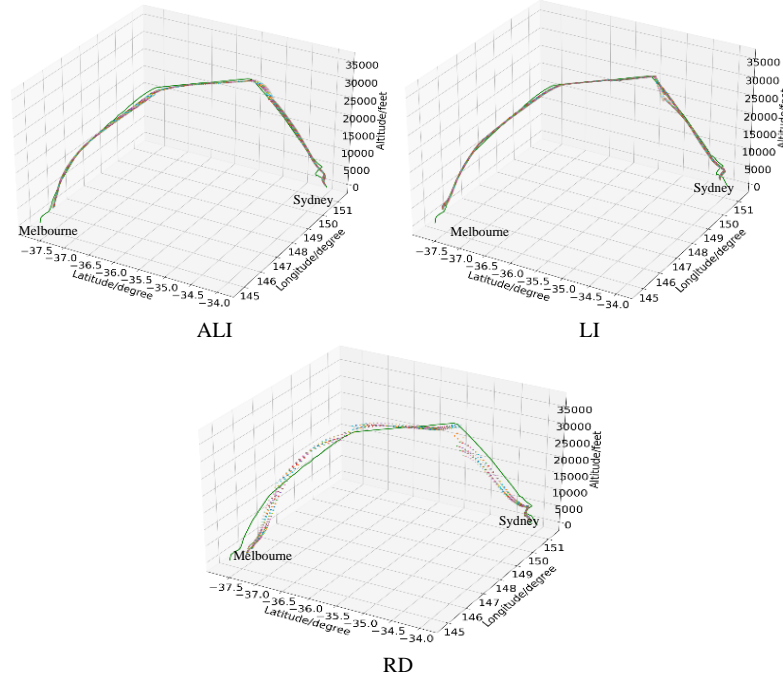


Fig. 5. Predicted flight trajectories.

While 3D trajectory plotting provides an intuitive representation, it does not allow for the quantification or display of errors based on timestamps. To conduct a more comprehensive analysis of the advantages and disadvantages, the table below presents the three different errors associated with the three pre-processing methods in the three directions for this prediction.

Table 1. Error for three pre-processing methods.

	MAE			MAPE			RMSE		
	ALI	LI	RD	ALI	LI	RD	ALI	LI	RD
X	0.061	0.031	0.084	1.6E-3	8.8E-4	2.4E-3	0.060	0.032	0.086
Y	0.078	0.023	0.14	5.3E-4	1.6E-4	9.2E-4	0.081	0.024	0.14
Z	328.2	306.8	904.0	0.023	0.025	0.104	342.6	328.8	957.4

When assessing overall accuracy, LI outperforms the other two algorithms. This is primarily because LI has access to the largest amount of training data. Additionally, LI maintains a constant time interval, which reduces data complexity. ALI exhibits better accuracy than RD, even though it shares a similar amount of data with RD. The key reason for this difference is that RD's data time interval is entirely unpredictable, while ALI follows a more structured data pattern that can be easily detected by LSTM. Analyzing individual axes, ALI demonstrates a significant advantage on the Z-axis. This can be attributed to the fact that ALI is based on the data variation rate along the Z-axis, making it easier for the model to identify data patterns in this dimension. Nevertheless, ALI still maintains a considerable advantage over LI on the X and Y axes. For a more comprehensive performance assessment, MAE errors along the trajectory are plotted in Fig. 6. Although the other two error metrics are calculated, they are not included in this paper due to page limitations. With ALI, errors on the X and Y axes are generally insignificant throughout the entire prediction. However, errors increase after 2200 seconds, likely due to an aircraft maneuver that results in less precise predictions.

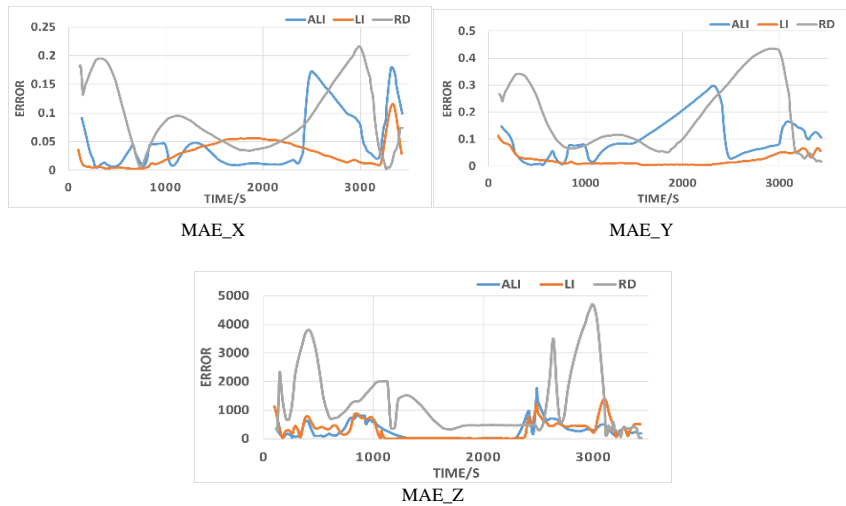


Fig. 6. Prediction error time-histories.

As shown in the figure, the ALI model demonstrates the ability to adjust accuracy and rapidly reduce data errors. When considering altitude, the model provides highly accurate predictions during specific time intervals which correspond to phases when the aircraft is climbing, descending, or cruising steadily. However, during other time periods, the altitude predictions are less accurate, particularly when the airplane is executing turns. In comparison, the RD model exhibits a similar error pattern for the X and Y axes. Nevertheless, it is evident that the RD model requires more time to adapt to abrupt changes in data. This can be attributed to two factors: first, the RD dataset contains less data compared to ALI or LI, and second, the time interval of raw data in RD is highly unpredictable, leading to increased errors. Conversely, LI con-

sistently demonstrates good accuracy, even during turning maneuvers, as its time interval remains constant. To assess whether the error patterns observed previously are consistent across different trajectories, the model underwent testing with 30 additional trajectories in this section. The average of all three errors was calculated and presented in Fig. 7. To ensure that all errors are plotted on the same scale, altitude units are adjusted to 1000 feet. The results indicate that the performance of X, Y, and Z predictions, as shown earlier, aligns with the universal performance across these trajectories, with minor variations. In terms of altitude, ALI exhibits slightly higher accuracy compared to LI. This difference may be attributed to the fact that LI treats all data as equally important by maintaining a constant time interval, whereas ALI assigns greater importance to highly variable data by increasing its data density through additional training. For other phases of the trajectory, where the data is less complex, the model's performance is less affected by data density, as the data density doesn't significantly impact accuracy in these cases. This study highlights that ALI has the capability to restructure the data density pattern effectively, catering to training requirements and substantially reducing training time by filtering out less crucial data.

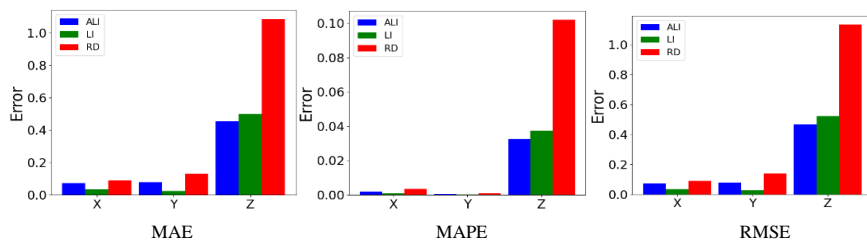


Fig. 7. Average prediction error.

5 Conclusion

This paper extensively discussed the Adjustable Interpolation Algorithm (ALI) and its impact on prediction accuracy when integrated into an LSTM model. ALI, as a variant of traditional interpolation methods, dynamically adjusts time intervals based on the ascent/descent rate of aircraft. The paper thoroughly examined its functionality by analyzing error metrics in three dimensions (longitude, latitude, and altitude) for ALI, Linear Interpolation (LI), and Raw Data (RD) through testing on individual and multiple trajectories. The findings revealed that ALI, while exhibiting slightly larger errors than LI in longitude and latitude, and similar errors in altitude, offers significant advantages in reducing training time. ALI is deemed highly suitable for applications in Urban Air Mobility (UAM) scenarios where extensive training is required. However, for commercial airspace operations where regular training is less essential, LI is recommended. Furthermore, ALI has the potential to aid Air Traffic Control Officers (ATCOs) during international flight handovers by providing timely information about the location and timing of handovers, enhancing situational awareness, and mitigating air traffic risks. For future research and development, the paper suggests exploring additional combinations of ALI time intervals to optimize the model setup. Alterna-

tive machine learning methods will be also explored to determine which ones can best accommodate ALI. Additionally, future work will also focus on low-altitude ATM applications and, in particular, flight in urban airspace.

6 Acknowledgements

This work was supported in part by the Hainan Provincial Natural Science Foundation of China Grant 720RC616, the National Natural Science Foundation of China Grant 62167003 and the Khalifa University Grant FSU-2022-013.

References

1. D. P. Thippavong *et al.*, "Urban air mobility airspace integration concepts and considerations," 2018 AIAA Aviation Technology, Integration, and Operations Conference, San Diego (CA), USA, 2018.
2. A. Straubinger, R. Rothfeld, M. Shamiyeh, K. D. Büchter, J. Kaiser, and K. O. Plötner, "An overview of current research and developments in urban air mobility – Setting the scene for UAM introduction," *Journal of Air Transport Management*, vol. 87, 2020, no. 101852.
3. S. Bijjahalli, A. Gardi, N. Pongsakornsathien, R. Sabatini, and T. Kistan, "A Unified Airspace Risk Management Framework for UAS Operations," *Drones*, vol. 6, 07/21 2022.
4. N. Pongsakornsathien *et al.*, "A Performance-Based Airspace Model for Unmanned Aircraft Systems Traffic Management," *Aerospace*, vol. 7, no. 11, p. 154, 2020.
5. L. Guan, J. Shi, D. Wang, H. Shao, Z. Chen, and D. Chu, "A trajectory prediction method based on bayonet importance encoding and bidirectional LSTM," *Expert Systems with Applications*, vol. 223, 2023, Art no. 119888.
6. I. Dhief, Z. Wang, M. Liang, S. Alam, M. Schultz, and D. Delahaye, Predicting Aircraft Landing Time in Extended-TMA using Machine Learning Methods. 2020.
7. W. Qin, J. Tang, and S. Lao, "DeepFR: A trajectory prediction model based on deep feature representation," *Information Sciences*, vol. 604, pp. 226-248, 2022/08/01/ 2022.
8. T. Chen *et al.*, "An Improved Multimodal Trajectory Prediction Method Based on Deep Inverse Reinforcement Learning," *Electronics*, vol. 11, no. 24, 2022, no. 4097.
9. Z. Chang, M. Hu, Y. Zhang, X. Zhu, and M. Li, "A 4D Trajectory Prediction Method for Aircraft under Wind," in *Proceedings of SPIE 2023*, vol. 12642.
10. Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based Flight Trajectory Prediction," in *2018 Int. Joint Conference on Neural Networks (IJCNN)*, 8-13 July 2018, pp. 1-8.
11. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *27th International Conference on Neural Information Processing Systems*, Montreal, Canada, 2014.
12. M. Gao, G. Shi, and S. Li, "Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network," *Sensors*, vol. 18, no. 12, 2018, no. 4211.
13. L. Mou, P. Ghamisi, and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639-3655, 2017.
14. F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *IEE Conference Publication*, 1999, vol. 2, 470 ed., pp. 850-855.