IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2021, accepted January 24, 2022, date of publication February 4, 2022, date of current version February 17, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149231

Aircraft Trajectory Prediction With Enriched Intent Using Encoder-Decoder Architecture

PHU N. TRAN, HOANG Q. V. NGUYEN, DUC-THINH PHAM, AND SAMEER ALAM
School of Mechanical and Aerospace Engineering, Air Traffic Management Research Institute, Nanyang Technological University, Singapore 639798

Corresponding author: Sameer Alam (sameeralam@ntu.edu.sg)

This work was supported in part by the National Research Foundation, Singapore; and in part by the Civil Aviation Authority of Singapore under the Aviation Transformation Program.

ABSTRACT Aircraft trajectory prediction is a challenging problem in air traffic control, especially for conflict detection. Traditional trajectory predictors require a variety of inputs such as flight-plans, aircraft performance models, meteorological forecasts, etc. Many of these data are subjected to environmental uncertainties. Further, limited information about such inputs, especially the lack of aircraft tactical intent, makes trajectory prediction a challenging task. In this work, we propose a deep learning model that performs trajectory prediction by modeling and incorporating aircraft tactical intent. The proposed model adopts the encoder-decoder architecture and makes use of the convolutional layer as well as Gated Recurrent Units (GRUs). The proposed model does not require explicit information about aircraft performance and wind data. Results demonstrate that the provision of enriched aircraft intent, together with appropriate model design, could improve the prediction error up to 30% at a prediction horizon of 10 minutes (from 4.9 nautical miles to 3.4 nautical miles). The model also guarantees the mean error growth rate with increasing look-ahead time to be lower than 0.2 nautical miles per minute. In addition, the model offers a very low variance in the prediction, which satisfies the variance-standard specified by EUROCONTROL (EU Organization for Safety and Navigation of Air Traffic) for trajectory predictors. The proposed model also outperforms the state-of-the-art trajectory prediction model, where the Root Mean Square Error (RMSE) is reduced from 0.0203 to 0.0018 for latitude prediction, and from 0.0482 to 0.0021 for longitude prediction in a single prediction step of 15 seconds look-ahead. We showed that the pre-trained model on ADS-B data maintains its high performance, in terms of cross-track and along-track errors, when being validated in the Bluesky Air Traffic Simulator. The proposed model would significantly improve the performance of conflict detection systems where such trajectory prediction models are needed.

INDEX TERMS Aircraft trajectory prediction, 4D trajectory, machine learning, encoder-decoder, convolution neural network, recurrent neural network.

I. INTRODUCTION

A. AIRCRAFT TRAJECTORY PREDICTION

Aircraft trajectory prediction is a crucial component of any air traffic control (ATC) system. *Aircraft trajectory* is defined as "a four dimensional (e.g., latitude, longitude, altitude and time) description of an aircraft's flight path" [1]. Trajectory prediction refers to the estimation of a flight's future trajectory within a look-ahead time (prediction horizon) [1]. Accurate aircraft trajectory prediction not only improves situational awareness of air traffic control officers (ATCOs), but also provides necessary inputs for other ATC functionalities such as departure and arrival management, monitoring aids, medium-term conflict detection (MTCD), short-term conflict alert (STCA), etc [2]. For example, improvement in the accuracy of trajectory prediction in the look-ahead time of approximately 4-8 minutes could potentially help to reduce STCA's nuisance alerts, which in turns enhances the system overall efficiency. In the Trajectory-Based Operations (TBO) concept [3], which is envisioned as the key enabler for the next generation of air traffic management, trajectory prediction is an essential building block for the Decision Support Tools (DST) for ATCOs.

Traditional aircraft trajectory predictors compute the aircraft future trajectories using diverse information such as flight-plan, aircraft performance, meteorological forecast, ATCOs and flight crews intents.

Major sources of errors in traditional approaches include diversity in aircraft performance, uncertainties in input data, ATCOs intents, and in longitudinal progress (Figure 1a) [4].

The associate editor coordinating the review of this manuscript and approving it for publication was M. Shamim Kaiser.

Prediction errors also resulted from limited accuracy of numerical integration algorithms and simplified aircraft's equations of motion [5]–[7]. Among the factors that hinder the trajectory prediction, uncertainties in the intents of ATCOs is recognized as the most significant one [4], which are influenced by the dynamics of traffic flow, weather conditions, operational constraints, etc. Figure 1b illustrates the flown trajectory of flights from Kuala Lumpur International Airport to Singapore Changi Airport. Although those flights were planned to follow a fixed route, there was a high variances in the realizations of the planned routes making trajectory prediction a challenging task.

In the past few decades, the rapid advances in machine learning algorithms have offered new approaches to improve the performance of computational models in solving classical problems in air traffic management [8]. Approaches to the aircraft trajectory prediction are also being transformed by the paradigm shift towards data-driven methods using machine learning techniques [9], [10]. As mentioned earlier, the lack of information about aircraft intent is a primary source of trajectory prediction errors. In this study, we aim to model and incorporate aircraft intent in a deep learning model for trajectory prediction. The aircraft intent used in our work is modeled from the historical flown trajectories of the aircraft, which in turn is derived from the Automatic Dependent-Surveillance Broadcast (ADS-B) data. In this work, we make use ADS-B data as the only data source, without other salient features such as wind and weather information. As for short-term trajectory prediction, the effect of wind and weather is considerably stable and is reflected in the actual track data, which can be learned by the proposed model as latent vectors. In the following section I-B a review of the related work is presented followed by details of research objectives and research contribution in Section I-C.

B. RELATED WORK

Aircraft trajectory prediction is specifically important to aircraft separation assurance, which includes conflict detection and resolution. Conflict detection algorithms can be classified into nominal, worst-case and probabilistic approaches [11], [12]. Among these, the probabilistic approach is a trade-off between detection accuracy and computational time required under various uncertainties. Probabilistic conflict detection requires as input the predicted positions of the aircraft with cross-track and along-track errors, by which one can estimate the probability of the distances (both vertical and lateral) between any two aircraft to be lower than the standard separations [13], [14]. Therefore, the estimation of trajectory prediction errors plays a key role in the provision of an accurate conflict detection. The increasing availability of aircraft historical trajectory data (e.g. ADS-B) and advanced techniques in trajectory mining and pattern analysis [15], [16] have been contributing to the enrichment of trajectory data, which allows further improvements of prediction models.

Various machine learning algorithms have been considered for aircraft trajectory prediction. Regression methods using ordinary least square and neural network were employed in [17] for the task of predicting altitude of the aircraft, which showed that both regression approaches perform significantly better than the point-mass model. In [18], the authors used Generalized Linear Model (GLM) to predict aircraft's arrival time at significant points, and the model can also be used for spacing the aircraft in continuous descent operation. Multiple regression models approach was also considered for trajectory prediction in [19], in which aircraft trajectories in Terminal Maneuvering Area (TMA) were classified by their distinct patterns, and different regression models were developed for different groups of trajectory to predict the aircraft arrival time. Such approach, i.e., trajectory clustering combined with multiple predictive models, were also employed for trajectory prediction in [20]–[23]. Other recent studies on the employment of different machine learning algorithms for trajectory prediction include Bayesian deep neural networks [24], [25], variational inference [26], conditional generative adversarial network [27], deep Gaussian process [28]. In addition, a hybrid machine learning-physics approach was recently proposed [29], in which an estimation algorithm (Residual-Mean Interacting Multiple Model) was introduced to improve a machine learning models by accounting for the motion of the aircraft.

Deep learning has improved the state-of-the-art in many problems that challenge the conventional machine learning methods such as speech recognition, visual object recognition, object detection. For processing sequential inputs, recurrent neural network (RNN) is often a better approach. However, traditional RNN has problems of learning long-term dependencies and vanishing gradient. Long Short Term Memory [30] (LSTM), a type of RNN, was introduced to overcome these problems and have been used to advance the state of the art for many difficult problems such as handwriting recognition and generation, language modeling and translation. A simpler variant of LSTM, Gated Recurrent Unit (GRU) [31] was introduced to deal with the vanishing problem and long-term depencies, which combines the forget and input gates into a single ''update grate,'' and provides the similar performance on certain tasks such as speech signal modeling or natural language processing, with LSTM. These RNN models can be used as modules inside encoder-decoder sequence-to-sequence architecture [32] for problems with both sequential input and output. This encoder-decoder architecture was used for solving human mobility trajectory prediction [33] or for the vehicle trajectory prediction problem [34] and was shown to outperform other non encoder-decoder methods. RNN was also combined with graph model in a structural-RNN framework in [35], which can learn the trajectory patterns of different agents (pedestrians, bicycles or cars). The approach based on 4D graph (two dimensions for instances and their interactions, one for time series and one for high-level categorization), and showed a improvement of 20% of accuracy over previous models. LSTM and attention mechanism [36] were integrated within an encoder-decoder architecture in [37] to perform

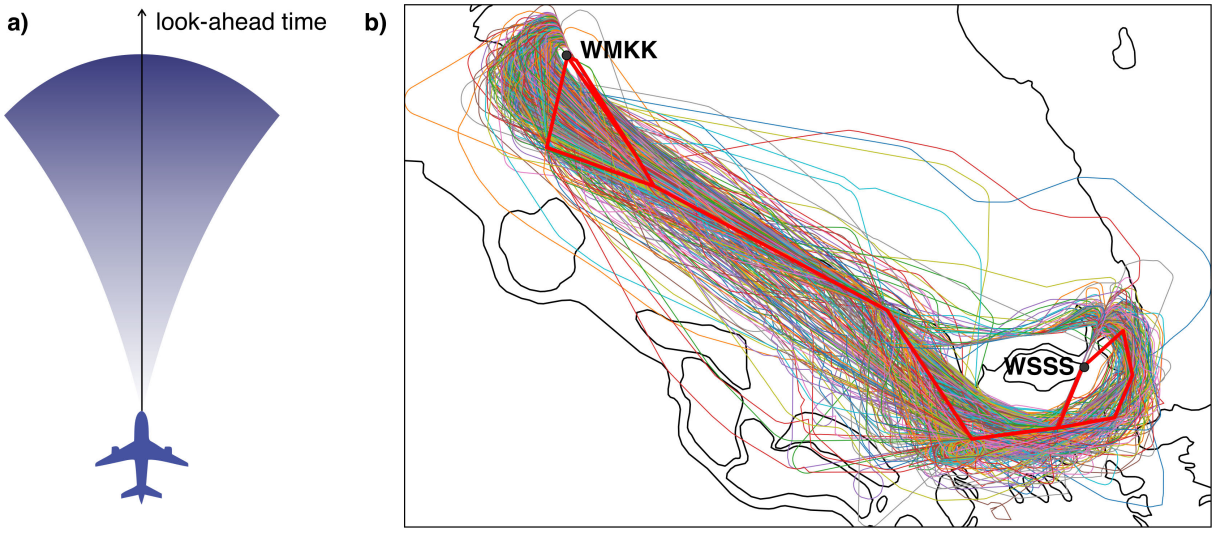


FIGURE 1. (a) Cone of uncertainty in aircraft future position. The color gradient indicates degree of uncertainty, darker color means higher uncertainty. Uncertainty in aircraft position increases with look-ahead time. (b) High variances in the actual flown tracks of flights from Kuala Lumpur International Airport (WMKK) to Singapore Changi Airport (WSSS). The thick red curves are planned routes.

geo-sensory time series prediction. The attention mechanism included local spatial, global spatial and temporal attentions (multi-level attention-based recurrent neural network), which outperformed nine baseline models. [38] proposed a Hierarchical Spatial-Temporal Long-Short Term Memory Network (HST-LSTM) for location prediction, in which a hierarchical extension was used for embedding the users' visiting context. The results showed the effectiveness of the approach on real-world dataset. A combination of attention mechanism and LSTM network for prediction of human location was proposed in [39], which extended the LSTM network with attention-based sequence-to-sequence for learning human semantic trajectories, which was shown to be beneficial to modelling semantic trajectories and prediction of human movement patterns. [40] proposed two variants of RNN, namely Constrained State Space RNN (CSSRNN) and Latent Prediction Information RNN (LPIRNN), that can incorporate unique trajectory constraints, of which normal LSTM models are incapable. The proposed variants showed slightly better results than that of ordinary LSTM networks.

As aircraft trajectories can be naturally modeled as time sequences of aircraft states, many studies employed sequence learning techniques, such as RNN, GRU, and LSTM, for trajectory prediction. A LSTM network was proposed in [41] for trajectory prediction and was shown to outperform other methods such as Markov Model and weighted Markov Model. GRU neural network was also employed for trajectory prediction in [42], which showed accuracy improvements compared to LSTM network. A combination of LSTM and convolutional layer was investigated in [43], in which the model can predict aircraft trajectory between any arbitrary two airports at low variance. Such combination of LSTM and convolutional layer was also employed in [44], where the prediction accuracy was shown to be increase by 21% comparing to using LSTM only. The LSTM network was also demonstrated to be superior to regression methods for trajectory prediction task [45]. Dynamic physical constraints was also taken into consideration within a LSTM network framework in [46], which showed the capability of integrating both long-term dependencies and dynamic constraints in trajectory prediction. Another emerging approach is the vision-based approach, in which aircraft trajectory prediction is achieved using convolutional LSTM on a series of traffic scenario images [47]. Such approach is benefited from the automated feature extraction offered by the convolutional layers and the input dimension is independent of the scenario complexity (e.g., the number of aircraft).

We observed that different approaches using deep learning, specifically recurrent neural network, had been recently employed for aircraft trajectory prediction, as those methods very well fit the spatio-temporal property of trajectory data. Nevertheless, there were limited work that give special consideration to model the aircraft intent as input features for deep learning model. Furthermore, some important evaluation metrics such as along-track error, cross-track error, and the error growth rates overtime had not been properly compared with available standard, such as the standard specified by EUROCONTROL [2]. In this work, we propose an approach overcoming these limitations.

C. OBJECTIVES & SUMMARY OF CONTRIBUTIONS

The objective of this research is to propose a machine learning model for aircraft trajectory prediction that takes into consideration the aircraft intent, in a prediction horizon of 1 to 10 minutes.

The proposed model adopts the Encoder-Decoder architecture [32], which can take a sequence as input and generate another sequence as output. This makes Encoder-Decoder architecture a suitable choice for trajectory prediction because a trajectory can be naturally expressed as a

time-sequence of spatial positions. Furthermore, the Encoder part of the model can extract latent features automatically, which in turns is used by the Decoder to produce a prediction.

The input to the model is the aircraft current trajectory, which is defined as a time sequence consisting of current position of the aircraft and its k last positions. The output of the model, or the predicted trajectory, is a time sequence of n positions of the aircraft into the future. Given that all trajectories have equal time interval between any two consecutive positions, n also determines the look-ahead time or the prediction horizon. Model architecture is described in detail in Section II.

In this research, the trajectory prediction is limited to en-route phase. The main contributions of this paper include:

- An effective way to model aircraft intent for the integration into the trajectory prediction model. Experiments show that our intent modeling allows the trajectory predictor to improve performance at different prediction horizons. To the best of our knowledge, such modeling of aircraft intent for deep learning prediction model has not been reported previously.
- An encoder network architecture that effectively learns patterns in the behavior of aircraft without the need of explicit information about the aircraft performance.
- A decoder network architecture that fuses information about aircraft future intent into the knowledge about its recent behavior, and effectively predicts future trajectory of the aircraft.
- A loss function which consists of two components, namely *position loss* and *dynamics stability loss*. The high accuracy of the prediction model is achieved by minimizing this loss.
- The prediction horizon (i.e., look-ahead time) is independent of network architecture and can be dynamically changed even after the model has been trained.
- The proposed model requires limited amount of information. In specific, only historical tracks of the aircraft (ADS-B data) is necessary for model training.

The aforementioned features enable the proposed model to outperform existing models using the same approach and achieves prediction accuracy that is comparable to the standard specified by EUROCONTROL for aircraft trajectory prediction [2]. The rest of this paper is organized as follows. Section II elaborates learning model for trajectory prediction. Data source used in this work and features engineering are presented in Section III. Experiment setup, results and discussion are presented in Section IV. Finally, conclusions are drawn in Section VI.

II. LEARNING MODEL

This section elaborates the learning model for aircraft trajectory prediction that incorporates the aircraft intent. An overview of the proposed model is illustrated in Figure 2. First, the approach for aircraft intent modeling is presented. Second, a description of the data and data preparation steps is provided. Next, the learning model architecture, with explanation of different loss functions to achieve the learning objectives is presented. Finally, hyper-parameters and special treatments to improve training robustness in the experiments is discussed.

A. AIRCRAFT INTENT MODELING

Aircraft intent refers to the list of waypoints that the aircraft is set to traverse. Intent modeling is to express the relative relationship between the aircraft current position and these waypoints in a form suitable for the learning model. This relative relationship is encoded in a 10-dimensional vector, which is referred to as ***intent vector*** $\mathbf{I}_t$, where the subscript t indicates the time step.

Figure 3 shows an aircraft at the current time in the context of three waypoints: previous, current, and next waypoints. The aircraft passed the previous waypoint, is passing by the current waypoint, and then heading to the next waypoint. An intent vector includes the four distances c_1 and c_2 ($c_1, c_2 \geq 0$), d_1 and d_2 ($d_1, d_2 \geq 0$) and the two angles β_1 and β_2 ($-\pi \leq \beta_1, \beta_2 < \pi$). Here, c_1 and c_2 are the cross-track distances from the aircraft to the current and the next airways, respectively. d_1 and d_2 are the distances from the aircraft to the current and the next waypoints, respectively. β_1 and β_2 are the angles measured from the instantaneous moving direction of the aircraft to the lines connecting the aircraft to the current and the next waypoints, respectively. The intent vector also includes the lateral directions from the aircraft to the current and the next waypoints. The ten elements of an intent vector are described in Table 1.

TABLE 1. Elements of the intent vector. See Figure 3 for a visual explanation.

Features	Dimension
$c_1, d_1, \cos\beta_1$	3
$c_2, d_2, \cos\beta_2$	3
Lateral direction from the aircraft to the current waypoint	2
Lateral direction from the aircraft to the next waypoint	2
Total length	**10**

Physically, d_1 and d_2 indicate the progress of actualizing the intent by the aircraft, while c_1 and c_2 imply the lateral deviation of the flown trajectory from the planned route. In addition, β_1 and β_2 observe the track angle evolution with respect to the three waypoints. These information helps the model to discriminate between turning and non-turning (maintaining heading) behaviors. The intent vector is dynamically computed during prediction to reflect the changes in the three waypoints as the aircraft is moving and passing them one by one. The computation of the intent vector is performed in a local Cartesian coordinates system (see Section III-C2). Note that the intent modeling is still valid if the current and the next airways form a straight line with intermediate waypoints.

B. THE PREDICTION TASK

Let $\mathbf{p}_i$ denotes the aircraft position at the discrete time step i, $\mathbf{p}_i = \left(p_i^{(1)}, p_i^{(2)}, p_i^{(3)}\right)$ where $p_i^{(1)}, p_i^{(2)}, p_i^{(3)}$ are the coordinates

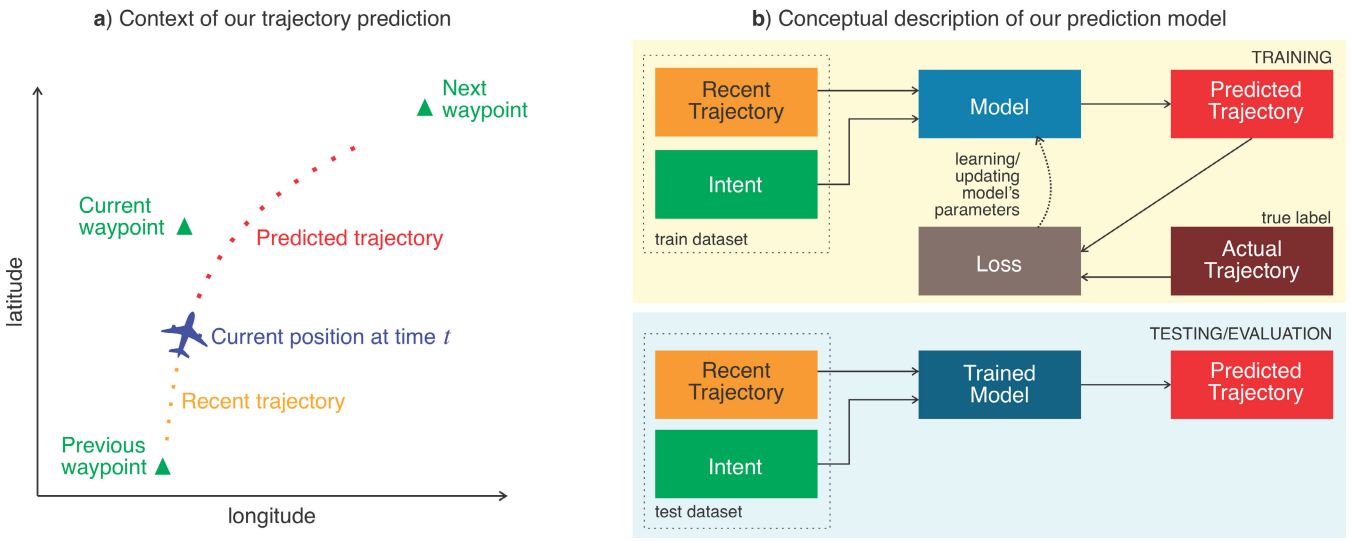


FIGURE 2. Overview of the learning model. (a) Context of trajectory prediction problem. The proposed model performs trajectory prediction in the spatial context of three waypoints (previous, current and next waypoints). Note that the altitude dimension is not included in the figure for a simple presentation. (b) Conceptual illustration of the prediction model. The model is trained using the train dataset (upper part) and the trained model performs predictions using the test dataset (lower part).

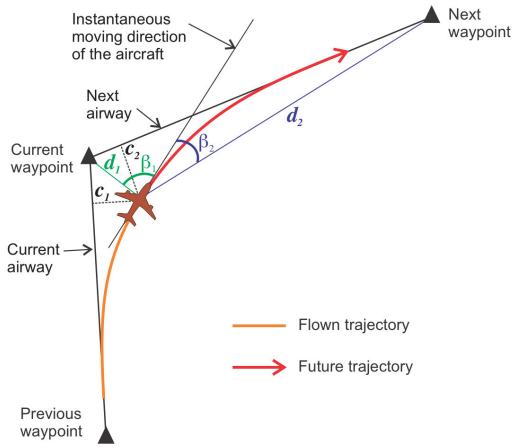


FIGURE 3. Aircraft intent modeling. Intent of the aircraft are encoded by the distances c_1, c_2, d_1, d_2 and the angles β_1, β_2, in the spatial context of three waypoints.

of the aircraft in a three-dimensional Cartesian reference frame. Assuming that $\mathbf{p}_t$ is the aircraft position at the current time t, the last k positions are $[\mathbf{p}_{t-k}, \mathbf{p}_{t-k+1}, \ldots, \mathbf{p}_{t-1}]$. The next n positions, i.e., future trajectory, of the aircraft is defined as $\mathcal{T}_t^n = [\mathbf{p}_{t+1}, \mathbf{p}_{t+2}, \ldots, \mathbf{p}_{t+n}]$ (Figure 2a). Given the current and the last k positions of the aircraft, the task is to predict its future trajectory. In other words, we train a model that receives a vector $[\mathbf{p}_{t-k}, \mathbf{p}_{t-k+1}, \ldots, \mathbf{p}_{t-1}, \mathbf{p}_t]$ as an input and produces a prediction $\hat{\mathcal{T}}_t^n = [\hat{\mathbf{p}}_{t+1}, \hat{\mathbf{p}}_{t+2}, \ldots, \hat{\mathbf{p}}_{t+n}]$ as the output. The model also receives intent vector as an input. The learning objective is to minimize the difference between the prediction $\hat{\mathcal{T}}_t^n$ and the actual trajectory $\mathcal{T}_t^n$. In the final evaluation settings, we set $k = 9$ and $n = 40$. This means the model receives an input sequence of 10 last positions (including the aircraft current position) and produces an output sequence of 40 positions.

C. MODEL ARCHITECTURE

The learning model adopts the encoder-decoder architecture [32], which particularly suits our problem for several reasons. First, the encoder-decoder architecture allows to conveniently handle the difference in length of the input and output data. In fact, the model takes as input a fixed-length vector of aircraft recent positions and produces a variable-length vector of aircraft future trajectory. Second, the internal structure of the encoder and that of the decoder are independent of each other. Thus, they can be designed to perform different tasks. As such, the encoder is trained to recognize hidden information and patterns in the behaviors of the aircraft. These knowledge is output by the encoder as a fixed-length context vector. After that, the decoder dynamically concatenates the context vector and the intent vector, and produces predicted trajectory. The overall architecture of our model is illustrated in Figure 4. Details about the encoder and the decoder are described below.

1) ENCODER

The encoder is designed to learn and recognize hidden patterns in the recent trajectory of the aircraft. Those hidden patterns carry information about aircraft dynamics characteristics under different circumstances, which are essential to the prediction of future trajectory. To achieve this, the encoder consists of one one-dimensional convolution (Conv1D) layer and one fully connected layer, each followed by a ReLU activation (Figure 4).

Convolutional layer makes use of convolution operator in place of general matrix multiplication [48, p. 224]. Some important features of convolutional layers are weights sharing and their ability to capture the local connectivity. These features make convolutional layers a natural choice

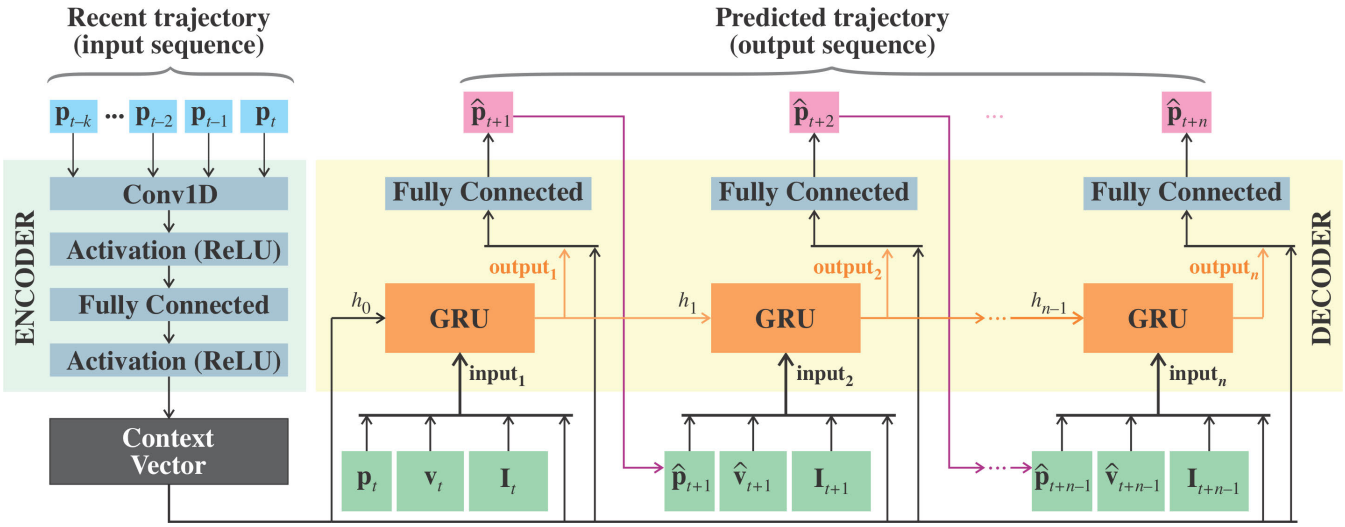


FIGURE 4. Encoder-decoder architecture for trajectory prediction. The decoder is unrolled for illustration of *n* sequential prediction steps (only three steps are visible in the figure).

for patterns recognition in time sequence data like aircraft trajectory.

The output from the encoder, which is defined as the context vector, has a length of l_c. Here, l_c is a hyper-parameter and determined experimentally. The context vector carries essential information about the current aircraft dynamics and other uncertainties such as wind effect. As the prediction horizon considered in this work is relatively short, changes in the wind field and other uncertain factors are insignificant. As a result, the learned information in the context vector can be used in the prediction of the aircraft future locations.

The parameters of the encoder are presented in Table 2.

2) DECODER

In the proposed model, the decoder is a Recurrent Neural Network (RNN). Particularly, RNN is a class of neural networks where connections between nodes form a directed graph along a sequence. RNN is designed to recognize sequential patterns for inferring the next likely outcome. Figure 5a shows a structure of a RNN. The vanilla RNN has issues of gradient vanishing and exploding when dealing with long-term series. *Gated Recurrent Unit* (GRU), a type of RNN, was introduced in [31] to overcome these issues in capturing long-term dependencies. The structure of a GRU cell is demonstrated in Figure 5b, and its internal computational logic is described by

$$\begin{cases} z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \\ r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \\ \hat{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right) \\ h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \end{cases} \tag{1}$$

where W_z, W_r, W are the coefficient matrices, σ is the sigmoid function and $*$ is the element-wise multiplication operator.

The decoder is designed to perform n-step sequential prediction. Figure 4 illustrates the unrolled GRU network of the decoder that predicts the future trajectory of n steps, $\hat{\mathcal{T}}_t^n = \left[\hat{\mathbf{p}}_{t+1}, \hat{\mathbf{p}}_{t+2}, \ldots, \hat{\mathbf{p}}_{t+n}\right]$. At each prediction step, in general, the GRU receives the current input and the previous hidden state, and produces a GRU output. This GRU output is then employed in two ways. First, it is concatenated with the context vector and then input to a fully connected layer in order to produce the aircraft position prediction for the current step. Second, it is used as the GRU hidden state for the subsequent prediction step (see Figure 4).

The input to the GRU at every step is illustrated in Figure 4. At a prediction step i, for instance, the GRU input is from the concatenation of four vectors:

- the last predicted position of the aircraft $\hat{\mathbf{p}}_{i-1}$,
- the last predicted velocity of the aircraft $\hat{\mathbf{v}}_{i-1} = \hat{\mathbf{p}}_{i-1} - \hat{\mathbf{p}}_{i-2}$,
- the current intent vector as described in Section II-A, and
- the context vector produced by the encoder.

The concatenation of these four vectors forms a GRU input vector of length $(l_c + 16)$, where l_c is the length of the context vector (Table 2). At the first prediction step, the GRU hidden state h_0 is initialized to the context vector. The details of decoder is given in Table 2.

TABLE 2. Parameter settings used in our experiment.

	Layers	Parameters
Encoder	Conv1D	kernel_size=3, output_channels=16
	Fully Connected	output=32 (l_c)
Decoder	GRU	input_size=48, hidden_size=32, num_layers=1
	Fully Connected	output=3 (ENU coordinate)

3) LOSS FUNCTION

The model is trained by minimizing the loss that has two components, namely *position loss* $\mathcal{L}_{\mathrm{P}}^t$ and *dynamics stability loss* $\mathcal{L}_{\mathrm{S}}^t$.

The position loss $\mathcal{L}_{\mathrm{P}}^t$ measures the deviation of the predicted trajectory $\hat{\mathcal{T}}_t^n$ from the actual one $\mathcal{T}_t^n$. It is determined

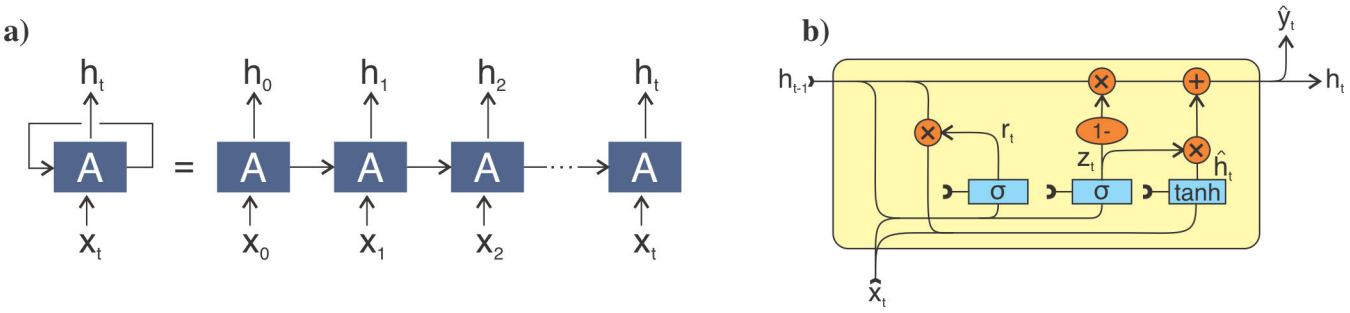


FIGURE 5. a) An unrolled recurrent neural network; b) Internal structure of a Gated Recurrent Unit [31].

by the RMSE of the Euclidean distances between corresponding points in $\hat{\mathcal{T}}_t^n$ and $\mathcal{T}_t^n$, i.e.,

$$\mathcal{L}_{\mathrm{P}}^t = \sqrt{\sum_{i=1}^{n} \mathrm{d}^2\left(\hat{\mathbf{p}}_{t+i}, \mathbf{p}_{t+i}\right)}, \tag{2}$$

where $\mathrm{d}(\cdot\ ,\cdot)$ yields the Euclidean distance between two positions.

The dynamics stability loss, as the name suggests, reflects the stability of the aircraft dynamics. It captures the variation in the *predicted* aircraft velocity. Let $\hat{\mathbf{v}}_i = \hat{\mathbf{p}}_i - \hat{\mathbf{p}}_{i-1}$ is the predicted velocity of the aircraft at a time step i. The dynamics stability loss of the n-step prediction horizon is determined by

$$\mathcal{L}_{\mathrm{S}}^t = \sqrt{\sum_{i=1}^{n} \mathrm{d}^2\left(\hat{\mathbf{v}}_{t+i}, \hat{\mathbf{v}}_{t+i-1}\right)} \tag{3}$$

Note that when $i = 1$, $\hat{\mathbf{v}}_{t+i-1} \equiv \hat{\mathbf{v}}_t = \mathbf{p}_t - \mathbf{p}_{t-1}$, which is the current velocity of the aircraft and can be determined from the model's input. The total loss is then defined as

$$\mathcal{L}^t = \mathcal{L}_{\mathrm{P}}^t + \alpha \mathcal{L}_{\mathrm{S}}^t \tag{4}$$

where α is a weight coefficient that determines the contribution of the dynamics stability loss to the total loss.

The inclusion of the dynamics stability loss is beneficial in several manners. First, minimizing this loss helps the model to learn the aircraft dynamics effectively. Physically, the dynamics stability loss measures the fluctuation in the predicted aircraft velocity in every time interval. As the trajectory predictor is designed for en-route phase, during which the aircraft velocity is relatively stable, it is desirable that this velocity fluctuation to be insignificant. Second, experiments shows that the second term in the right-hand side of Equation 4 has similar effect of a regularization term in typical context of machine learning. That means, at an optimal value of α, minimizing the dynamics stability loss contributes to reducing over-fitting and improving generalization of the model. Here, α is treated as a hyper-parameter that needs to be determined empirically.

Let Θ denotes the set of all trainable parameters of the model. The learning objective is to determine the optimal set of parameters Θ^* that minimizes the total loss:

$$\Theta^* = \arg\min_{\Theta} \mathcal{L} = \arg\min_{\Theta} \sum_{\mathcal{T}_t} \left(\mathcal{L}_{\mathrm{P}}^t + \alpha \mathcal{L}_{\mathrm{S}}^t\right) \tag{5}$$

III. DATA SOURCE & FEATURES ENGINEERING

In this section, we present the data source and features engineering, including data pre-processing steps and aircraft intent reconstruction from the trajectory data.

A. INTENT RECONSTRUCTION

As described in Section II-A, aircraft intent refers the planned route of the aircraft. In this work, only ADS-B data is used, without using filed flightplans of the flights presented in the data. Thus, we reconstruct the planned routes, or the intent, of the flights from their flown track data.

Intent reconstruction is performed at two levels. On the first level, given a set of all designated significant points, we identify those points that are sufficiently close to the flown track of a flight, and consider them as the planned route. Here, designated significant points (or waypoints) are specified geographical locations in the airspace used to define flight routes, and they can be collected from the Electronic Aeronautical Information Publication (eAIP) of the related Flight Information Regions (FIRs). For convenience, we refer to the intent on this first level as ***ordinary intent***.

ATCOs often vectors aircraft from their original plans due to reasons such as conflict resolution, congestion management, flows optimization, etc. In those situations, however, the ATC clearances (in the form of voice instructions) given to the flight crews are not captured in the ADS-B data. On the second level of intent reconstruction, we also identify missing trajectory change points (i.e., turn location) that were resulted from ATC clearances, in addition to the designated significant waypoints. We refer to the intent on this second level as ***enriched intent***. Note that to facilitate the enriched intent, we first need to identify all turn segments in the trajectory data.

1) TURN SEGMENT IDENTIFICATION

We determine the turn segments, noted as $\mathcal{S}_t$, and non-turn segment, $\mathcal{S}_s$, by the geometric curvature and along a trajectory. Recall that the curvature of a curve κ and its relationship

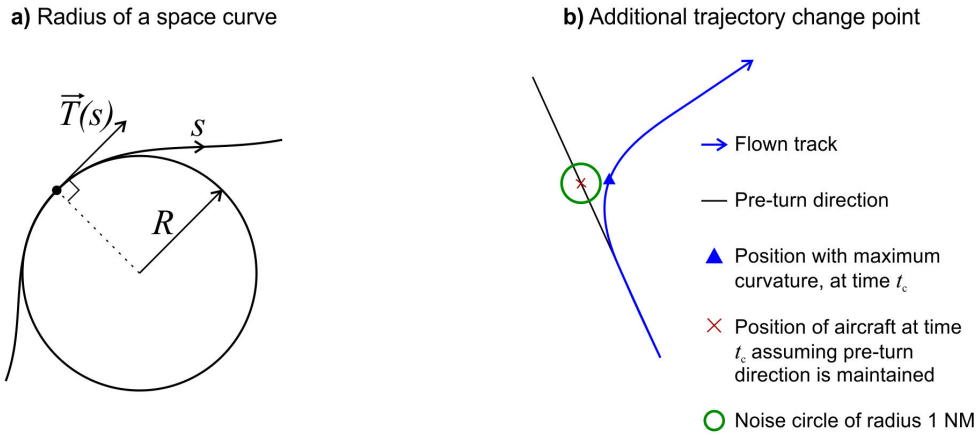


FIGURE 6. (a) Radius of a space curve in the calculation of curvature. (b) The determination of additional trajectory change point for enriched intent. The addition trajectory change point is randomly located in the green circle.

with the curve radius R are given by:

$$\kappa = \left|\frac{d\vec{T}}{ds}\right| = \frac{1}{R} \tag{6}$$

where $\vec{T}$ is the unit tangent vector of the trajectory, s is the arc length, and R is the radius (Figure 6a). We define a turn segment as a trajectory segment where all the points in the segment have $R \leq 40$ NM. This radius threshold was determined empirically to reduce turns mis-detection.

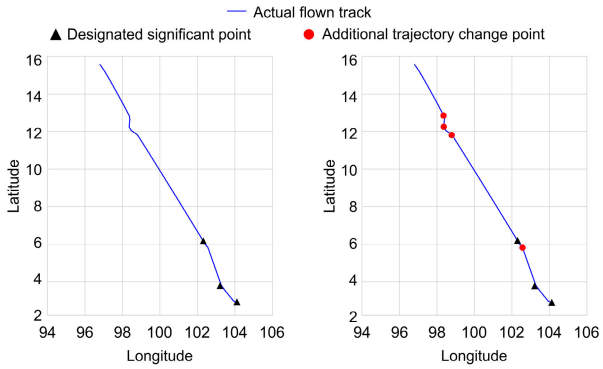


FIGURE 7. Example of reconstructed ordinary intent (left) and enriched intent (right) for a flown track.

2) ORDINARY AND ENRICHED INTENTS RECONSTRUCTION

Intent reconstruction is performed following the steps below.

a: ORDINARY INTENT

We choose designated significant waypoints to include in the flightplan of a flight. For non-turn segment, $\mathcal{S}_s$, a waypoint is included when it is less than 2 NM apart from the flight's flown track. For turn segment, $\mathcal{S}_t$, this threshold is 7 NM. The ordinary intent for a trajectory can be written as:

$$\mathcal{I}_O = \{\mathbf{wp}_i \in \mathcal{W} \,|\, dis(\mathbf{wp}_i, \mathcal{S}_s) \leq 2 \text{ NM} \text{ or } dis(\mathbf{wp}_i, \mathcal{S}_t) \leq 7 \text{ NM}\} \tag{7}$$

where $dis(\mathbf{wp}_i, \mathcal{S})$ is the closest Euclidean distance from waypoint $\mathbf{wp}_i$ to a segment $\mathcal{S}$, and $\mathcal{W}$ is the set of all waypoints in the airspace.

b: ENRICHED INTENT

To identify an additional trajectory change point (Figure 6b), first, we determine the time t_c at which the curvature reaches its maximum value during the turn. Second, we find the position of the aircraft at t_c assuming the aircraft was following the pre-turn direction. Finally, we randomly choose a point within a noise circle of radius 1 NM, whose center is placed at the position found in the second step (the green circle in Figure 6b). The enriched intent for a trajectory can be expressed by:

$$\mathcal{I}_E = \mathcal{I}_O \cup \{TCP_i\} \tag{8}$$

where each TCP_i is an identified trajectory change point. An example of ordinary and enriched intents for a trajectory is demonstrated in Figure 7.

B. DATA SOURCE AND PRE-PROCESSING

The data used in this work is Automatic Dependent Surveillance–Broadcast (ADS–B) provided by the Civil Aviation Authority of Singapore. The data covers a geographical region of latitude from N0°0′ to N25°0′, longitude from E85°0′ to E125°0′ (Figure 8), and dated from March 15th 2019 to April 30th 2019. We consider all the aircraft flown tracks in the mentioned region during the chosen time window, and this results in a total of 16,884 flight trajectories. The following pre-processing steps were applied.

1) DATA FILTERING AND TRAJECTORY RE-SAMPLING

The following conditions were applied for filtering data to remove noises:

- A trajectory needs to have at least one hour of data in the mention region and has no missing positions during a time window of 1000 seconds.

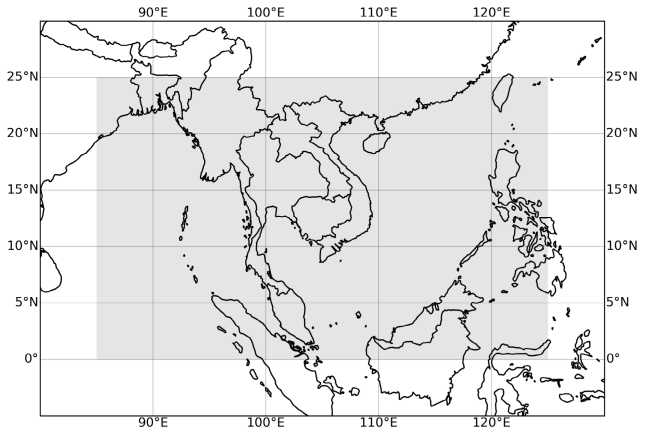


FIGURE 8. The geographical region (shaded area) of the ADS-B data considered in this work.

- The computed ground speed from aircraft's positions should be stable. Therefore, we discard trajectories that give rise to speed outliers. Concretely, the top 1 percent and the bottom 1 percent in distribution of computed speeds is omitted.
- Only parts of the trajectories that are between flight levels 290 and 410 are maintained since we focus on the en-route phase.

After this filtering process, 2882 trajectories remained for training and evaluation. Since the proposed model requires the input and output sequences to have equal time interval between any two consecutive data points, we employed cubic spline interpolation for trajectory re-sampling with a fixed time interval of 15 seconds. We chose this re-sampling interval because it helps to reduce the amount of training data being fed to the model and therefore speed up model training without compromising the prediction accuracy.

2) TRAIN & TEST DATASETS SPLITTING

We construct the train dataset using the trajectory data dated from March 15th to March 25th, and the remaining data is used for testing and evaluation. The train dataset includes 690 flight tracks with 358,294 data points and the test dataset includes 2192 flight tracks with 1,130,913 data points. We use data from the first 11 days, about 25% of the total data, to train the model and use the remaining data to test the performance of the trained model. Experiments, which is discussed in detail in Section IV-A, showed that the proposed model converges with this amount of training data and there is little benefit of using more training data. We also observed that allocating more data for testing could help to better evaluate the prediction robustness and stability of the model.

C. DATASET PREPARATION

1) TRAINING SAMPLES GENERATION

From the pre-processed trajectory data, we generate a sample by choosing at random a point in a trajectory as the aircraft current position. We then combine the aircraft current position with k most recent positions to form the input sequence of length $(k+1)$. The sequence of next n points after the aircraft current position is regarded as the true value of the output sequence. Thus, each sample contains an input sequence and a true output sequence.

2) COORDINATES TRANSFORMATION

Initial experiments showed that the training is more robust when all positional coordinates in each training sample are expressed in a local Cartesian coordinate system instead of using the global geodetic coordinates (i.e., latitude, longitude, altitude). Therefore, for each sample, we transform all geographic positions from the global geodetic coordinate to a local tangent plane, i.e., east-north-up (ENU), coordinate. This transformation also applies to the computation of the intent vector. The origin of the local ENU frame is always placed at the aircraft current position of each data sample. The output of the decoder are also coordinates in the local ENU frame.

TABLE 3. Hyper-parameters settings for model training.

Parameters	Values
Input sequence length, k+1	10
Output sequence length, n	Random in range [5, 40]
Scheduler	Adam optimizer
Learning rate	0.0001
Dynamics stability loss coefficient, α	0.25
Interval between 2 steps	15 seconds

IV. EXPERIMENTS & RESULTS

A. MODEL TRAINING AND HYPER-PARAMETERS

The proposed model architecture and learning algorithm are implemented in Python language using PyTorch deep learning library. In particular, we utilize a mini-batch stochastic gradient descent method to train our network model with automatic differentiation. The Adam optimizer, a variant of this method implemented in PyTorch, is selected to use in our experiment with learning rate set to 0.0001.

RNNs in general and GRUs specifically are slow to train due to the fact that elements in the output are computed sequentially, one after another. Using large mini-batch size and taking advantage of GPU computing can reduce the training time significantly. To further save the training time, we start with large mini-batch size (i.e., 256 samples per batch) at the beginning of the training, and gradually reduce the batch size to as small as 32 samples. In such manner, the final set of model's parameters are obtained after 100 training epochs.

In the final settings, our model predicts the next 40 future positions of the aircraft given the most recent 10 positions, including the current position. In the training phase, however, the output length of the decoder is not necessarily fixed. In fact, we observed that the model converges faster and more robust if we allow the output length to alternatively take

values from the range [5, 40]. More specifically, small values of n (i.e., 5 or 10) help the model to converge faster in the "right" direction at the beginning. After that, larger values of n allow the model to improve its performance on longer prediction horizon. Thus, in training time, n takes controlled random values within this range, as indicated in Table 3. Once the model has been trained, we set $n = 40$ for evaluation. Such flexibility in the output length is offered by the GRU network of the decoder.

We also empirically determine the dynamics stability loss coefficient (α) to be 0.25. Hyper-parameters regarding the model architecture are shown in Table 2. Hyper-parameters for model training are indicated in Table 3.

B. INDEX OF PERFORMANCE

Euclidean distance is a common choice for measuring similarities in trajectory prediction evaluation. We employ Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the error in the predicted trajectory.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \text{d}^2\left(\hat{\mathbf{p}}_i, \mathbf{p}_i\right)} \tag{9}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \text{d}\left(\hat{\mathbf{p}}_i, \mathbf{p}_i\right) \tag{10}$$

where n is the length of output sequence, $\hat{\mathbf{p}}_i$ and $\mathbf{p}_i$ are the true and predicted positions, respectively.

We also consider along-track error (ATE), cross-track error (CTE) for performance evaluation [5]:

$$ATE = \Delta p^{(1)} \sin\varphi + \Delta p^{(2)} \cos\varphi \tag{11}$$

$$CTE = \Delta p^{(1)} \cos\varphi - \Delta p^{(2)} \sin\varphi \tag{12}$$

where $\Delta p^{(1)} = \hat{p}^{(1)} - p^{(1)}$, $\Delta p^{(2)} = \hat{p}^{(2)} - p^{(2)}$, φ is the angle between the aircraft track and the north direction. Note that $p^{(1)}$ and $p^{(2)}$ are coordinates in the local ENU reference frame. The ATE and CTE are illustrated in Figure 9.

C. RESULTS ANALYSIS

Table 4 compares model's performance on inputs with ordinary intent and that with enriched intent, in terms of RMSE and MAE of the Euclidean distances between true positions and the predicted ones, at different look-ahead times. In general, the prediction errors grow linearly with the prediction horizon. When ordinary intent is provided, from 1 to 10 minutes look-ahead time, RMSE grows from 0.49 NM to 4.9 NM, and MAE from 0.3 NM to 3.3 NM. When enriched intent is available, RMSE ranges from 0.4 NM to 3.4 NM and MAE from 0.3 to 2.6 NM, from 1 to 10 minutes of prediction horizon. One can also observe that the RMSE/MAE ratio in the case with ordinary intent is higher than that when using enriched intent. This suggests that large errors arise more frequently when the provided intent information is insufficient or incorrect, which happens when the intent is not enriched.

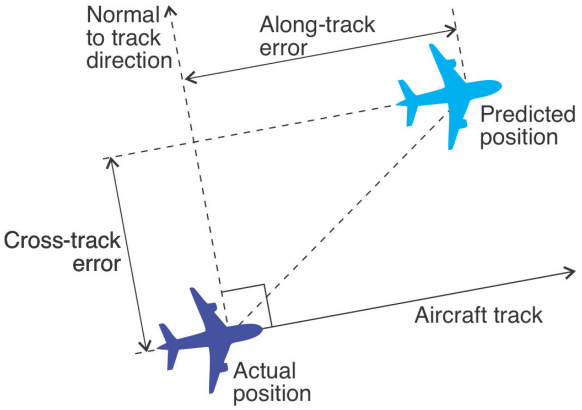


FIGURE 9. Along-track error (ATE) and cross-track error (CTE) of predicted aircraft position.

The analysis in Table 4 also indicates that providing the model with enriched intent significantly reduces both errors, comparing to the case where only ordinary intent is considered. In specific, when enriched intent is used, RMSE improves approximately 14% to 30% depending on the look-ahead time, and MAE improves approximately 11% to 20%. It is very interesting to observe that the improvement in two types of error is more significant when the prediction horizon is farther into the future. This suggests that enriched intent of the aircraft plays a more important role in maintaining accurate prediction when one performs prediction with longer look-ahead time. For short-term prediction, the recent state of the aircraft influences the prediction performance more than the intent does. In fact, finding the dynamic balance between the influence of recent behavior and that of future intent when making prediction at different look-ahead times is an important exercise in model design.

TABLE 4. Model prediction performance in terms of RMSE and MAE (in nautical miles) at different look-ahead times. The results are shown for two different levels of intent reconstruction.

	Ordinary Intent		Enriched Intent		Percentage Improvement	
Look-ahead time	RMSE	MAE	RMSE	MAE	% RMSE	% MAE
1 min	0.490	0.354	0.421	0.316	14.08	10.73
2 mins	0.996	0.690	0.818	0.601	17.87	12.90
3 mins	1.505	1.020	1.181	0.868	21.53	14.90
4 mins	2.017	1.355	1.527	1.133	24.29	16.38
5 mins	2.523	1.690	1.858	1.393	26.36	17.57
6 mins	3.024	2.026	2.182	1.652	27.84	18.46
7 mins	3.513	2.359	2.497	1.907	28.92	19.16
8 mins	3.989	2.691	2.808	2.162	29.61	19.66
9 mins	4.453	3.023	3.120	2.421	29.93	19.91
10 mins	4.911	3.361	3.441	2.689	29.93	19.99

We now examine further the model performance by investigating cross-track error (CTE) and along-track error (ATE). As shown in Figure 9, CTE indicates how much the aircraft slide off the actual course, while ATE measures the difference between the projected position along the actual course and its

true value. Figure 10 presents boxplots of these two errors at different prediction horizons. For CTE (Figure 10a), the model achieves a good prediction accuracy, where 50% of the errors are below 1 NM and 75 % of them below 1.8 NM, at 10 minutes prediction horizon. For ATE (Figure 10b), the median value of the error goes from about 0.15 NM at 1 minute look-ahead time to 1.5 NM at 10 minutes. The ATE is not as low as the CTE due to the fact that uncertainty grows faster during the longitudinal progress. One attribute of the ATE is that it can be used to estimate the error in arrival time of the aircraft at certain point in the future, given information about the aircraft speed. A common property of CTE and ATE observed from Figure 10 is that they both linearly increase with the look-ahead time. The accumulation of errors over many sequential prediction steps can not be completely avoided, unfortunately. Nevertheless, the model is successful in avoiding large explosion of the errors over time, and the consistent linear relationships between errors and prediction horizon makes the errors' behavior more predictable.

Figure 11 shows the distributions of CTE and ATE at three different prediction horizons of 2, 5, and 10 minutes, performed on the test dataset. These distributions also indicate that a majority of test cases produce low errors. For example, it can be observed that at 2 minutes look-ahead time, about 75% of the cases have both CTE and ATE below 0.5 NM. When we look farther into the future, e.g., at 5 minutes, about 75% of the test cases have CTE below 1 NM and have ATE below 1.3 NM. Longer look-ahead time amplifies the gap between CTE and ATE because of the effect of increasing uncertainty during the longitudinal progress, as we mentioned earlier.

The model predicts the aircraft future trajectory without information about aircraft performance. In air traffic management, aircraft performance (or aircraft type) is usually associated with their wake turbulence categories (WTC), i.e., light, medium, and heavy aircraft. To verify the model performance for different aircraft categories, a further investigation on the prediction errors is performed on different WTCs. Figure 12a shows the statistics of different aircraft types presented in the test dataset, and how they are grouped into medium and heavy WTCs (en-route airspace have medium and heavy aircraft only). We allow the model to perform prediction for the medium and the heavy groups separately, and the results are shown in Figure 12b in terms of mean Euclidean distance errors. It is observable from Figure 12b that the difference in prediction errors between the two groups of WTCs is insignificant. The insignificant difference between these errors suggests that the model does not have any bias toward any specific WTC, and the model's performance is consistent regardless of aircraft WTC. This also indicates that our design of the encoder is able to characterize the dynamics of different aircraft types; thus, the explicit information of the aircraft type can be safely removed from the input. This is an important feature offered by our model as it helps to limit the amount of information needed while maintaining stable and high performance.

Now, we offer a comparison between performance of our model and that of the model recently developed in [46]. We choose the work [46] for benchmark for it also attempted to embed intent of the aircraft in the prediction model in the form of physical constraints. The model in [46] also made use of LSTM network, which can be considered as a variant of recurrent neural network like GRU. One difference between the two is that the model in [46] makes use of the constraint as a part of the loss function, while we input the aircraft intent to the decoder. In this study, the proposed model takes as input the last 10 positions and can predict future trajectory as a positions sequence of flexible length, where the time interval between two consecutive points is 15 seconds. The model in [46] takes as input the last 10 positions and predicts a position at 15 seconds into the future. Thus, for comparison, we allow our model to predict one step of 15 seconds and compare the prediction with that of [46]. The prediction error resulted from our model is accumulated after every prediction step; thus, a small error in a single-step prediction is desirable.

TABLE 5. Performance benchmark with state-of-the-art model.

	MAE		RMSE	
Models	latitude	longitude	latitude	longitude
cLSTM [46]	0.0050	0.0105	0.0203	0.0482
Model in this study	**0.0013**	**0.0016**	**0.0018**	**0.0021**

The comparison is presented in Table 5. The benchmark shows that our model performs significantly better and more stable than cLSTM does, in terms of MAE and RMSE of latitude and longitude predictions. The fact that the ratios RMSE/MAE of our model (1.38 for latitude and 1.31 for longitude) are significant lower than that of cLSTM (4.06 for latitude and 4.59 for longitude) suggests that our model produces very low variance compared to cLSTM, and that large errors happen less frequently in our case than they do in cLSTM.

TABLE 6. Model prediction increment in mean and standard deviation of errors by minute.

	Along-Track Error Growth Rate (NM/min)		Cross-Track Error Growth Rate (NM/min)	
Look-ahead time	Mean	Std	Mean	Std
1 min	+0.201	+0.167	+0.169	+0.161
2 mins	+0.167	+0.137	+0.164	+0.171
3 mins	+0.164	+0.132	+0.147	+0.160
4 mins	+0.176	+0.136	+0.135	+0.144
5 mins	+0.176	+0.136	+0.130	+0.134
6 mins	+0.179	+0.140	+0.126	+0.128
7 mins	+0.179	+0.139	+0.123	+0.121
8 mins	+0.179	+0.136	+0.122	+0.121
9 mins	+0.181	+0.134	+0.123	+0.127
10 mins	+0.190	+0.139	+0.125	+0.133

To further assess the model performance, we compute the error growth rates at different prediction horizons. The error growth rate measures how much the prediction error is worsened after every minute of look-ahead time. A good predictor should be able to keep these error growth rates within bounds. Table 6 presents the growth rates of ATE and

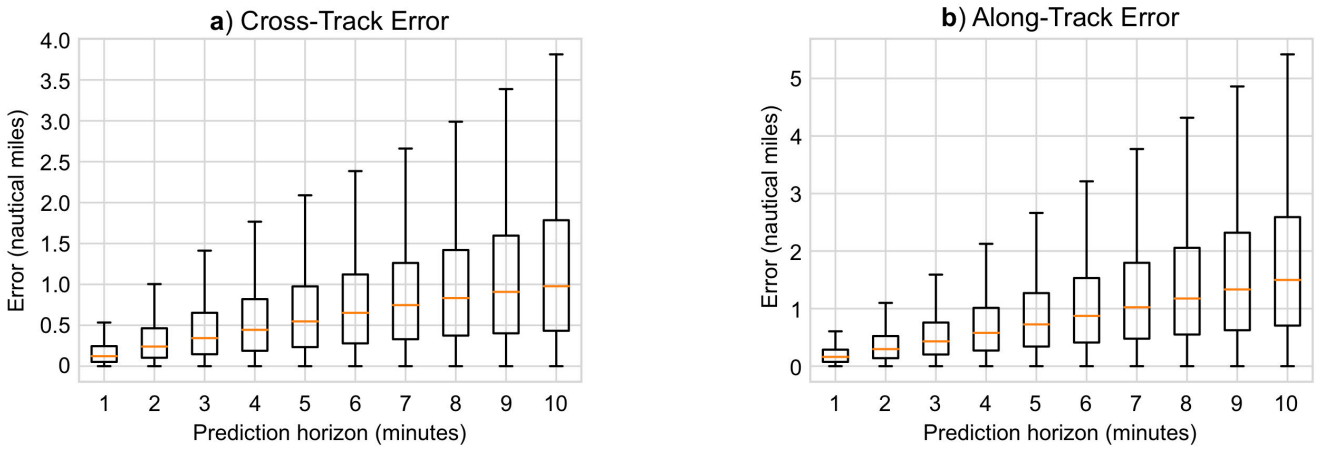


FIGURE 10. Boxplots of (a) Cross-Track error and (b) Along-Track error at different prediction horizons.

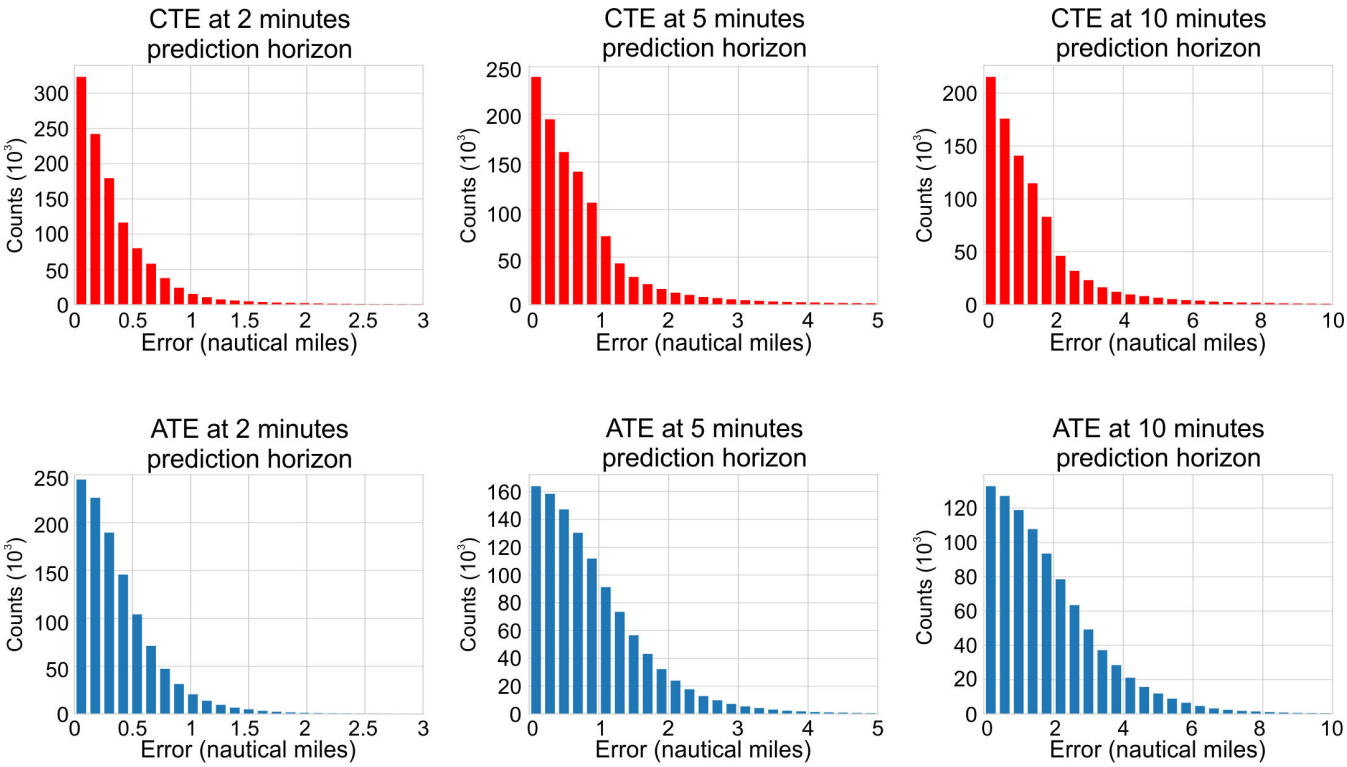


FIGURE 11. Distribution of Cross-Track Error (CTE) and Along-Track Error (ATE) at different prediction horizons. The first row shows CTE and the second row shows ATE. The three columns indicate three prediction horizons at 2, 5, 10 minutes, from left to right.

CTE in terms of mean and standard deviation (std) of the errors, in the unit of NM per minute. It can be observed that the error growth rates produced by our model are very stable, in terms of mean and std of the error, for both along-track and cross-track. This indicates that our model can effectively manage the accumulated prediction errors when prediction horizon becomes farther into the future.

The *EUROCONTROL Specification for Trajectory Prediction* [2] specifies the required growth rates for ATE of mean 0.1 NM/min and std 0.2 NM/min, and for CTE of mean 0.1 NM/min and standard deviation of 0.3 NM/min. In terms of mean growth rate, the model performance is quite close to the specifications. The model demonstrates that it can maintain a very low variances in the error growth rates despite increasing look-ahead time. In specific, for 10 minutes look-ahead, the proposed model offers ATE growth rate std ranging from 0.132 NM/min to 0.167 NM/min, which is lower than 0.2 NM/min. For CTE, in 10 minutes prediction horizon, the proposed model produces growth rates std ranging from 0.121 NM/min to 0.171 NM/min, which is significantly

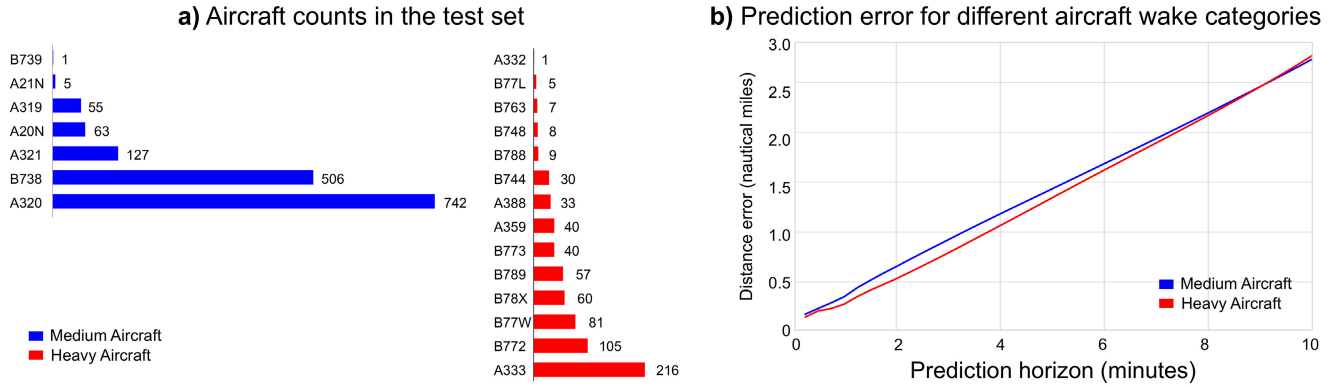


FIGURE 12. Prediction errors for different aircraft wake categories. (a) Number of different aircraft types in the test set. (b) Mean of prediction errors for medium and heavy aircraft.

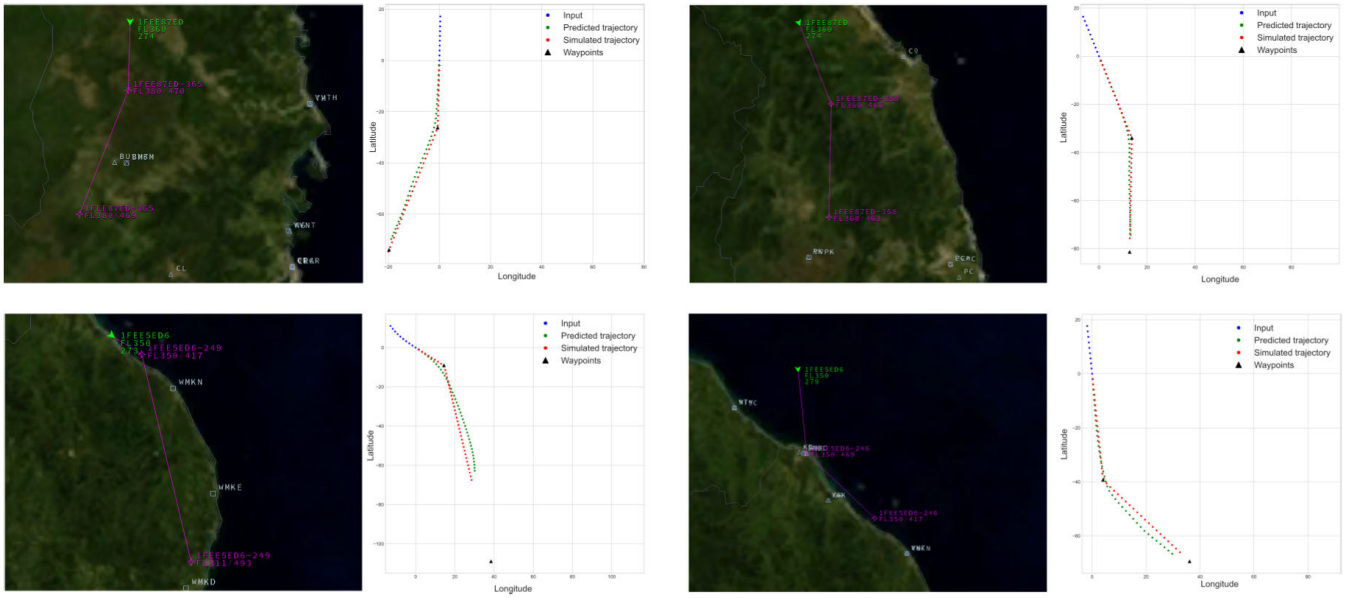


FIGURE 13. Four scenarios of aircraft trajectories being simulated in the BlueSky ATC Simulator and their corresponding predicted trajectories by the model. In each scenario, the left figure demonstrates the trajectory segment on the BlueSky's screen, and the right figure shows comparison between simulated (red) and predicted (green) trajectories, given the input to the model (i.e., recent locations of the aircraft) in blue.

low comparing to 0.3 NM/min. Although there is room for improvement in the mean of error growth rates, the results are promising if one takes into consideration that the model requires only recent trajectory and simple intent information as input.

V. SIMULATION VALIDATION

To further evaluate the performance of the prediction model, we perform model validation in an air traffic simulator. The BlueSky ATC Simulator [49] is used for this purpose. BlueSky is an open-source and research-oriented air traffic simulator that can integrate highly realistic aircraft dynamics via the EUROCONTROL's Base of Aircraft Data (BADA) [50]. To simulate an aircraft trajectory, the BlueSky simulator takes the flight-plan and aircraft type as inputs and generates a 4D trajectory. A flight-plan generally includes all the waypoints that the aircraft is going to traverse and the constraints (i.e., air speed, altitude, heading etc.) at each waypoint.

To facilitate the validation, we prepare a validation trajectories dataset in which each trajectory is simulated by the BlueSky's Flight Management System (FMS) given the respective pre-defined flight-plan. During flight simulation, locations of aircraft are recorded at every 15 seconds. The trained model takes the 10 most recent locations of an aircraft as inputs and predicts the aircraft's location in the next 10 minutes. Model performance is evaluated by the errors when comparing the predicted trajectories with the BlueSky simulated ones, in terms of along-track and cross-track errors. The validation dataset consists of 1,138,343 data points in total.

Figure 13 depicts four scenarios of aircraft trajectories being simulated in the BlueSky simulator and the comparison between the simulated trajectories and the predicted

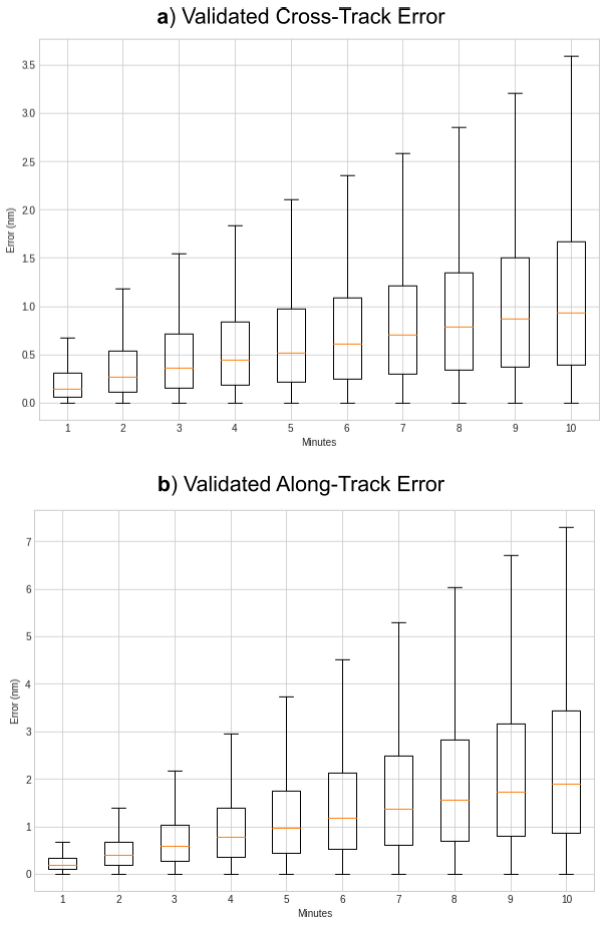


FIGURE 14. Boxplots of (a) cross-track and (b) along-track errors at different prediction horizons between the predicted trajectories by the model and the simulated trajectories by the BlueSky Simulator.

TABLE 7. Validation results in BlueSky simulator: along-track and cross-track errors growth rate at different prediction horizons.

	Along-Track Error Growth Rate (NM/min)		Cross-Track Error Growth Rate (NM/min)	
Look-ahead time	Mean	Std	Mean	Std
1 min	+0.245	+0.230	+0.253	+0.236
2 mins	+0.250	+0.228	+0.203	+0.277
3 mins	+0.251	+0.235	+0.136	+0.166
4 mins	+0.249	+0.234	+0.103	+0.122
5 mins	+0.250	+0.233	+0.095	+0.107
6 mins	+0.252	+0.234	+0.098	+0.101
7 mins	+0.252	+0.232	+0.103	+0.101
8 mins	+0.247	+0.225	+0.109	+0.109
9 mins	+0.234	+0.217	+0.114	+0.121
10 mins	+0.213	+0.203	+0.118	+0.132

ones. These four scenarios are chosen to demonstrate the model's performance at different kinds of aircraft intent (i.e., waypoints in the flight-plan) and at different segments (i.e., straight and turn segments) of the trajectory. It is observable that the model predicts the aircraft future locations equally well during the straight and turn segments of the trajectory. ATE and CTE of the validation are presented in Figure 14, in which the errors resulted from the validation in BlueSky Simulator are comparable with the errors performed on the ADS-B test dataset (Figure 10). Furthermore, the cross-track and the along-track error growth rates during simulation validation, as shown in Table 7, also approach the test results performed on ADS-B data earlier (Table 6). This consistency in model's performance suggests that the model, which is trained on ADS-B dataset, can generalize well in the simulation environment without the need of further training.

VI. CONCLUSION

In this work, a deep learning model, based on encoder-decoder architecture, is proposed for aircraft trajectory prediction by modeling and incorporating aircraft intent. Specifically, an encoder network that effectively learns patterns in the behavior of aircraft without the need of explicit information about the aircraft performance and a decoder network that fuses information about aircraft future intent into the knowledge about its recent behavior, and effectively predicts future trajectory of the aircraft is designed. We demonstrate that an effective modeling and incorporation of intent, could improve the prediction performance up to 30% in terms of Root Mean Squared Error of Euclidean distance between predicted and true positions, at 10 minutes prediction horizon (look ahead time for Conflict Detection models in Air Traffic Control systems). Our approach also produce very low variance in the prediction, compared to the standards required by EUROCONTROL. The accumulated prediction error over time is also well managed by the proposed model. The model is also able to discriminate different aircraft dynamics during prediction without the need of explicit information about aircraft wake categories. With these features, the proposed model outperforms the existing state-of-the-art model in aircraft trajectory prediction. Another benefit from the proposed approach is that the model requires minimal amount of information to perform the prediction.

A key factor that contributes to the safe operation of an airspace sector is the situational awareness of air traffic controllers, and this in turn depends on how well the controllers anticipate the traffic movement, particularly all aircraft future locations. Thus, an accurate and reliable aircraft trajectory prediction would be valuable for the controllers to probe any potential incidents in the sector and resolve them timely and efficiently. Future extensions of this work may include (1) extended design of model architecture for incorporation of multi-aircraft intent, (2) modeling of uncertainty in the implementation of ATCO instructions by flight crews, (3) analysis of effects of traffic flow management strategies used by ATCOs on aircraft intent, (4) further investigation of model performance using actual ATC clearance data instead of enriched intent, etc.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore. *(Phu N. Tran and Hoang Q. V. Nguyen contributed equally to this work.)*

REFERENCES

[1] (2004). FAA/EUROCONTROL. *Action Plan 16: Common Trajectory Prediction Capability*. Accessed: Sep. 18, 2020. [Online]. Available: https://sites.google.com/site/trajectorymgt/ap16—white-paper

[2] (2017). EUROCONTROL. *EUROCONTROL Specification of Trajectory Prediction*. Accessed: Sep. 18, 2020. [Online]. Available: https://www.eurocontrol.int/publication/eurocontrol-specification-trajectory-prediction

[3] ICAO. *ICAO Global TBO Concept*. Accessed: Dec. 15, 2021. [Online]. Available: https://www.icao.int/airnavigation/tbo/PublishingImages/Pages/Why-Global-TBO-Concept/

[4] S. Swierstra and S. Green, "Common trajectory prediction capability for decision support tools," in *Proc. 5th USA/Eurocontrol ATM R&D Seminar*, Budapest, Hungary, 2003, pp. 23–27.

[5] S. Mondoloni, S. Swierstra, and M. Paglione, "Assessing trajectory prediction performance-metrics definition," in *Proc. 24th Digit. Avionics Syst. Conf.*, vol. 1, Oct. 2005, p. 3.

[6] M. Paglione and R. Oaks, "Implementation and metrics for a trajectory prediction validation methodology," in *Proc. AIAA Guid., Navigat. Control Conf. Exhib.*, Aug. 2007, p. 6517.

[7] S. Torres, "Determination and ranking of trajectory accuracy factors," in *Proc. 29th Digit. Avionics Syst. Conf.*, Oct. 2010, p. 1.

[8] B. Sridhar, "Applications of machine learning techniques to aviation operations: Promises and challenges," in *Proc. Int. Conf. Artif. Intell. Data Analytics Air Transp. (AIDA-AT)*, Feb. 2020, pp. 1–12.

[9] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: A review," *Big Data Mining Anal.*, vol. 1, no. 2, pp. 108–127, 2018.

[10] H. Georgiou, S. Karagiorgou, Y. Kontoulis, N. Pelekis, P. Petrou, D. Scarlatti, and Y. Theodoridis, "Moving objects analytics: Survey on future location & trajectory prediction methods," 2018, *arXiv:1807.04639*.

[11] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 179–189, Dec. 2000.

[12] J. Tang, "Conflict detection and resolution for civil aviation: A literature survey," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 10, pp. 20–35, Oct. 2019.

[13] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 199–220, Dec. 2000.

[14] W. Liu and I. Hwang, "Probabilistic trajectory prediction and conflict detection for air traffic control," *J. Guid., Control, Dyn.*, vol. 34, no. 6, pp. 1779–1789, 2011.

[15] Z. Yu, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.

[16] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2007, pp. 330–339.

[17] M. G. Hamed, D. Gianazza, M. Serrurier, and N. Durand, "Statistical prediction of aircraft trajectory: Regression methods vs point-mass model," ENAC, Toulouse, France, Tech. Rep. hal-00911709, 2013.

[18] A. de Leege, M. van Paassen, and M. Mulder, "A machine learning approach to trajectory prediction," in *Proc. AIAA Guid., Navigat., Control (GNC) Conf.*, Aug. 2013, p. 4782.

[19] S. Hong and K. Lee, "Trajectory prediction for vectored area navigation arrivals," *J. Aerosp. Inf. Syst.*, vol. 12, no. 7, pp. 490–502, 2015.

[20] T.-H. Le, P. N. Tran, D.-T. Pham, M. Schultz, and S. Alam, "Short-term trajectory prediction using generative machine learning methods," in *Proc. Conf. (ICRAT)*, Tampa, FL, USA, Sep. 2020, pp. 1–8.

[21] Z. Wang, M. Liang, and D. Delahaye, "Short-term 4D trajectory prediction using machine learning methods," ENAC, Toulouse, France, Tech. Rep. hal-01652041, Tech. Rep., 2017.

[22] G. Wang, H. Chen, K. Liu, R. Guo, and Y. Wei, "A flight trajectory prediction method based on trajectory clustering," in *Proc. IEEE 1st Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Oct. 2019, pp. 654–660.

[23] H. Georgiou, N. Pelekis, S. Sideridis, D. Scarlatti, and Y. Theodoridis, "Semantic-aware aircraft trajectory prediction using flight plans," *Int. J. Data Sci. Anal.*, vol. 9, no. 2, pp. 215–228, Mar. 2020.

[24] X. Zhang and S. Mahadevan, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decis. Support Syst.*, vol. 131, Apr. 2020, Art. no. 113246.

[25] Y. Pang, X. Zhao, H. Yan, and Y. Liu, "Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103326.

[26] Y. Pang and Y. Liu, "Probabilistic aircraft trajectory prediction considering weather uncertainties using dropout as Bayesian approximate variational inference," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 1413.

[27] Y. Pang and Y. Liu, "Conditional generative adversarial networks (CGAN) for aircraft trajectory prediction considering weather effects," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 1853.

[28] Z. Chen, D. Guo, and Y. Lin, "A deep Gaussian process-based flight trajectory prediction approach and its application on conflict detection," *Algorithms*, vol. 13, no. 11, p. 293, Nov. 2020.

[29] H.-C. Choi, C. Deng, and I. Hwang, "Hybrid machine learning and estimation-based flight trajectory prediction in terminal airspace," *IEEE Access*, vol. 9, pp. 151186–151197, 2021.

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

[31] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[32] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[33] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101441–101452, 2019.

[34] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1672–1678.

[35] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "TrafficPredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6120–6127.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[37] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," in *Proc. IJCAI*, 2018, pp. 3428–3434.

[38] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction," in *Proc. IJCAI*, vol. 18, no. 7, 2018, pp. 2341–2347.

[39] A. Karatzoglou, A. Jablonski, and M. Beigl, "A Seq2Seq learning approach for modeling semantic trajectories and predicting the next location," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 528–531.

[40] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, "Modeling trajectories with recurrent neural networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1–9.

[41] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[42] P. Han, W. Wang, Q. Shi, and J. Yang, "Real-time short-term trajectory prediction based on gru neural network," in *Proc. IEEE/AIAA 38th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2019, pp. 1–8.

[43] Y. Pang, N. Xu, and Y. Liu, "Aircraft trajectory prediction using LSTM neural network with embedded convolutional layer," in *Annu. Conf. PHM Soc.*, vol. 11, no. 1, pp. 1–10, 2019.

[44] L. Ma and S. Tian, "A hybrid CNN-LSTM model for aircraft 4D trajectory prediction," *IEEE Access*, vol. 8, pp. 134668–134680, 2020.

[45] P. Han, J. Yue, C. Fang, Q. Shi, and J. Yang, "Short-term 4D trajectory prediction based on LSTM neural network," *Proc. SPIE*, vol. 11427, Jan. 2020, Art. no. 114270M.

[46] Z. Shi, M. Xu, and Q. Pan, "4-D flight trajectory prediction with constrained LSTM network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7242–7255, Nov. 2021.

[47] H. Kim and K. Lee, "Air traffic prediction as a video prediction problem using convolutional LSTM and autoencoder," *Aerospace*, vol. 8, no. 10, p. 301, Oct. 2021.

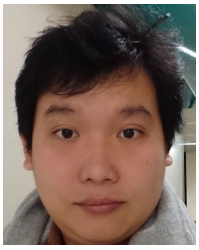
[48] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1, no. 2. Cambridge, MA, USA: MIT Press, 2016.

[49] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: An open data and open source approach," in *Proc. 7th Int. Conf. Res. Air Transp.*, vol. 131, 2016, p. 132.

[50] A. Nuic, "User manual for the base of aircraft data (BADA) revision 3.10," *Atmosphere*, vol 2010, p. 1, Feb. 2010.

PHU N. TRAN received the Ph.D. degree in mechanical engineering from Nanyang Technological University, Singapore, in 2017. He is currently a Postdoctoral Research Fellow at the Air Traffic Management Research Institute, Nanyang Technological University. His research interests include artificial intelligence, machine learning, and the applications of those techniques in air traffic control.

HOANG Q. V. NGUYEN received the M.Eng. degree in computer science from Télécom ParisTech, France, in 2018. He is currently a Research Associate at the Air Traffic Management Research Institute, Nanyang Technological University. His research interests include machine learning and data drive approaches and theirs application on trajectory prediction, conflict detection, and resolution for air traffic management.



DUC-THINH PHAM received the M.Sc. degree in computer science from Télécom ParisTech, France, in 2013, and the Ph.D. degree in computer science from Paris Sciences and Letters (PSL), France, in 2019. He is currently a Research Follow at the Air Traffic Management Research Institute, Nanyang Technological University, where he is also the Assistant Program Director of Artificial Intelligence and Data Analytics for Air Traffic Management Program and the Leader of data-driven surface movement management project. His research interests include on deep learning, deep reinforcement learning, and application of machine learning in air traffic management challenges, such as surface movement management and conflict detection and resolution.

SAMEER ALAM received the Ph.D. degree in computer science with specialization in artificial intelligence from the University of New South Wales (UNSW), Australia, in 2008. He is currently an Associate Professor at the School of Mechanical and Aerospace Engineering, Nanyang Technological University (NTU), Singapore, where he is also the Deputy Director of the Air Traffic Management Research Institute (ATMRI). His research interests include machine learning, computer vision, multi-agent systems, applied to air traffic, and airport operations. He is an Editorial Board Member of *Transportation Research Part C: Emerging Technologies*.