Transportation Research Part C 130 (2021) 103326

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach

Yutian Pang [a], Xinyu Zhao [b], Hao Yan [b], Yongming Liu [a,*]

[a] School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, 85287, AZ, USA
[b] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, 85287, AZ, USA

ARTICLE INFO

Keywords:
Trajectory prediction
Convective weather
Bayesian deep learning
Air traffic management

ABSTRACT

Trajectory prediction is an essential component of the next generation national air transportation system. Reliable trajectory prediction models need to consider uncertainties coming from multiple sources. Environmental factor is one of the most significant reasons affecting trajectory prediction models and is the focus of this study. This paper propose an advanced Bayesian Deep Learning method for aircraft trajectory prediction considering weather impacts. A brief review of both deterministic and probabilistic trajectory prediction methods is given, with a specific focus on learning-based methods. Next, a deterministic trajectory prediction model with classical deep learning methods is proposed to handle both spatial and temporal information using a nested convolution neural network, recurrent neural network, and fully-connected neural network. Following this, the deterministic neural network model is extended to be a Bayesian deep learning model to consider uncertainties where the posterior distributions of parameters are estimated with variational inference for enhanced efficiency. Both mean prediction and confidence intervals are obtained giving the last on-file flight plans and weather data in the region. The proposed methodology is validated using air traffic and weather data from the Sherlock data warehouse. Data pre-processing procedures for big data analytics are discussed in detail. Demonstration and metrics-based validation are performed during severe convective weather conditions for several air traffic control centers. The results show a significant reduction in prediction variance. A comparison with existing methods is also performed. Several conclusions and future works are given based on the proposed study.

1. Introduction

In the context of air traffic management (ATM), the development of the next generation national air transportation system (NextGen) (FAA, 2013) is of critical importance. NextGen aims to efficiently and safely accommodate the increasing air traffic flows within the United States. A key element of NextGen is the capability to predict and share air traffic trajectories and conflicts with all involved stakeholders. In NextGen, information sharing between aircraft will be greatly enhanced so that each aircraft receives and transmits the cooperative surveillance information. Thus, aircraft can take over a certain amount of ATM tasks from ground air traffic controllers. These decision-support tools (DSTs) (Schuster and Ochieng, 2014) include flight plan (FP) changes, dynamic weather reroute (DWR), trajectory prediction (TP), and conflict detection and resolution (CDR). With the advancement of these tools, NextGen can alleviate the mental workloads of ground air traffic controllers (Liu and Hwang, 2011; Phillips, 1996; Paielli and Erzberger, 1997), especially when infrequent but critical events happened (Sridhar et al., 1998).

* Corresponding author.
E-mail address: Yongming.liu@asu.edu (Y. Liu).

https://doi.org/10.1016/j.trc.2021.103326
Received 3 November 2020; Received in revised form 6 May 2021; Accepted 22 July 2021
Available online 2 August 2021

It has been shown that there are huge trajectory uncertainties during aviation operations. The uncertainty of trajectory predictions (TP) comes from multiple sources. A typical TP tool makes decisions based on the aircraft's performance, pilot's intent, and Terminal Radar Approach Control (TRACON) regulations (Thipphavong, 2008). The unawareness of pilot's intent and assumptions of aircraft's conditions contribute to the prediction uncertainty. Also, when multiple aircraft are heading to the same regions, the communication and avoiding maneuver between each of the aircraft will cause the uncertainty of the trajectory prediction (Knorr and Walter, 2011). Among all sources of uncertainties, the environmental factor is one of the most significant contributors to the TP uncertainty (Nilim et al., 2001; Lee et al., 2009; Mondoloni, 2006). The convective weather condition usually generates rapidly and randomly, which represents aleatoric uncertainties. The accuracy of the weather prediction, such as resolution, accuracy, and forecasting intervals, will also affect the TP uncertainties and represents epistemic uncertainty sources. In addition, the pilot and controller's decision for weather conditions brings in additional uncertainties from human factors. Due to the nature of uncertainties that contribute to the TP errors, a deterministic TP model will not be sufficient for the efficient and reliable TP models, especially when dealing with increasingly congested airspace (Ayhan and Samet, 2016).

The availability of different massive aviation databases (Arneson, 2018; Jones and Endsley, 1996) has enabled the possibility of changing the conventional TP tool into a multi-source data-driven approach (Zhang et al., 2011), such as machine learning (ML) methods. Deep learning (DL), as a special type of ML technique, draws significant interest in the academia and industries and has achieved great success in object detection (He et al., 2017), natural language processing (Manning et al., 1999; Collobert et al., 2011), dimension reduction (Hong et al., 2015), generative learning (Goodfellow et al., 2014), and reinforcement learning (Mnih et al., 2015). Numerous ML studies have been performed in ATM during the last decades, although multiple challenges (i.e., data collection, data privacy, data storage, data cleansing, and data opening) still exist in the real-world applications (Zhu et al., 2018). Compared to the traditional ML approaches, the DL model has the following advantages: (1) DL model has an architecture by stacking multiple back-propagation compatible layers to discover the inherent information within the data, such as the convolutional layers and recurrent layers to consider the complex spatial structures and temporal structures presented in the TP problem; (2) DL model can take advantage of the big data as the DL algorithm is easy to incorporate multiple factors into the TP model. However, the traditional DL model has the following limitations. (1) The inherent characteristics of DL models are prone to overfitting, especially for recurrent neural networks (RNNs) (Gal and Ghahramani, 2016a). To solve this, significant fine-tuning of network parameters combined with the stochastic regularization techniques (SRTs) such as dropout (Srivastava et al., 2014) are often required. Researchers typically choose the parameters of the DL model subjectively. (2) Traditional DL models cannot be used to quantify the uncertainties of the trajectories.

Bayesian techniques are known as a remedy to address the issues of overfitting and uncertainty quantification. More specifically, the DL model that incorporates Bayesian variational inference, known as Bayesian deep learning (BDL), is shown to be robust for overfitting problems (Lauret et al., 2008). Furthermore, instead of using a single point estimate of model parameters in classical DL, the Bayesian methods yield distribution for each of the parameters, which provides an inherent estimate of prediction uncertainty.

Therefore, we propose a Bayesian deep learning-based probabilistic aircraft trajectory prediction method under weather impact. Multiple convolutional, recurrent, and fully-connect layers are constructed to discover the complex spatial–temporal relationship from historical data. Bayesian theory is then applied to form a probabilistic learning framework. We perform our experiment with real flight trajectories, flight plans, and convective weather measurement data from the Sherlock database. To the best knowledge of the authors, this is the first paper that the BDL is used to perform the TP task considering weather impacts. Another new contribution of the proposed study is the proposed unique probabilistic CNN+RNN+FCNN architecture for big data analytics involving dynamically changing weather and trajectory information in ATM.

The rest of the paper is organized as follows. Section 2 reviews the studies performed in the general field of trajectory prediction. Section 3 discusses the evolution of deep learning combining Bayesian theory as well as our proposed trajectory prediction framework. Section 4 introduce the proposed TP framework as well as the processing pipeline on each of the three building-blocks. The test results of the proposed framework and the comparison with other NN models are described in Section 5. A study on the contributions of explicitly introducing weather features into the TP framework is also conducted. Concluding remarks and future directions are described in Section 6.

2. Literature review

Trajectory prediction has been long regarded as a key functional component in different research areas such as ground vehicle trajectory prediction (Wiest et al., 2012; Houenou et al., 2013; Ammoun and Nashashibi, 2009; Kim et al., 2017), and pedestrian trajectory prediction (Alahi et al., 2016; Xue et al., 2018; Gupta et al., 2018; Ziebart et al., 2009) in both deterministic and probabilistic sense. Here, we focus on the aircraft TP in the field of ATM. Existing methodologies can be divided into deterministic and probabilistic methods. Details are shown below.

2.1. Deterministic methods

The deterministic approach gives a point estimate of the aircraft location without associated uncertainties to the predicted trajectory. Most approaches focus on the modeling of aircraft's intent, motions, and dynamics. The nominal approach to do the prediction is to propagate an estimation of the state space and find the future predictions given a sequence of past observations.

2.1.1. Data-based TP

Most data-driven methods include state-space model (Ryan and Paglione, 2008), Kalman and particle filtering (Lymperopoulos and Lygeros, 2010), classical neural networks (NNs) (Le Fablec and Alliot, 1999), and physical-based learning (Yu et al., 2019). A state-space model is proposed for the short-time trajectory prediction near the terminal area by using a selection of flight plan or dead reckoning method instead of a flight plan (FP) based guidance (Ryan and Paglione, 2008). The key idea is that, if an aircraft is not following its route, the best estimation is using the dead reckoning. The experiment is conducted with four selected trajectories and evaluated by the distance between the predicted tracks and the radar recorded tracks. Some of the developed TP tools have been bringing into practice. A TP function for tactical flight management is developed based on aircraft motions (Benavides et al., 2014). These aircraft motions are simply an extension of aircraft states. This function continuously generates four-dimensional (4D) coordinates and updates the flight management system. The model has been deployed in the Advanced Concepts Flight Simulator at NASA Ames Research Center. Similar work on TP evaluates the future positions of the aircraft based on the aircraft maneuver state (Avanzini, 2004). Kalman filtering is used for state estimation and noise smoothing. The experiment is demonstrated by a computer simulation of reverse turn maneuvers of an aircraft model. Neural network has also been used in ATM. A vertical plan TP model is proposed using a simple neural network (Le Fablec and Alliot, 1999). The prediction is made with the knowledge of previous aircraft locations. The experimental result shows that NNs are more efficient and have lower average error comparing to non-parametric methods. Another group of deterministic approaches focuses on the selection of multiple trajectory candidates. These methods utilize the first part of the radar measurements of the flight tracks as the indicator for selection among trajectory candidates for single aircraft (Lymperopoulos et al., 2006) or multi-aircrafts TP (Lymperopoulos and Lygeros, 2010).

2.1.2. Human factors on TP

There is another research direction focusing on the human factors on TP. The pilot's intent is inferred and combined with aircraft motions and states for the TP task with a hybrid estimation algorithm (Yepes et al., 2007). This work first acquires the air traffic control (ATC) regulations, the flight plan, and the environmental factors to infer the pilot's intent. Then the prediction is modeled as a function of aircraft state and pilot's intent. The experiment is conducted with a 6-DOF (degree of freedom) simulation of Boeing 747-200 aircraft. Another interesting work is trying to model the aircraft trajectory into a set of intent-related languages (Besada et al., 2013). These languages are used for describing aircraft motions. A hierarchy of language generating engines is modeled and each engine can perform modification over a trajectory at different levels. Another recent work focuses on improving the accuracy of TP during landing by including voice communications into the Bayesian framework (Wang et al., 2021).

2.1.3. Issues with deterministic TP

The major issue of the deterministic approach for TP is the lack of ability to predict the uncertainty of the trajectory. There has been stated that the deterministic techniques suffers from the degraded accuracy in several cases (Benavides et al., 2014; Bronsvoort et al., 2012; Avanzini, 2004; Ryan and Paglione, 2008). Changing into probabilistic TP is a feasible solution to improve the robustness of the models. Furthermore, most of the trajectory prediction algorithms focuses on the short-term prediction, which cannot be used to predict the future trajectory in the long-term horizon.

2.2. Probabilistic methods

The probabilistic TP research experienced a transition from conventional methods to statistical ML-based methods. The conventional methods include Sequential Monte Carlo (SMC) (Crisostomi et al., 2008; Casado et al., 2012), Hidden Markov Model (HMM) (Qiao et al., 2014), and others (Swierstra and Green, 2003; Mueller et al., 2002; Prandini et al., 2000). ML-based methods are brought into the TP field due to the availability of surveillance data and the increased computing power. ML-based methods include the Gaussian Mixture Model (GMM), recurrent neural network (RNN), and several other statistical generalized regression models (GLMs).

2.2.1. Conventional TP

Probabilistic Monte Carlo (MC) methods and deterministic worst-case methods are proposed to perform model-based TP through a case study (Crisostomi et al., 2008) based on current observations. The uncertainty of this method is assumed to come from the inexact knowledge of wind and aircraft mass, which represents the aleatoric uncertainty. Similarly, another research tries to identify the source of uncertainty that affecting a 3-DOF point mass model of aircraft motion (Casado et al., 2012). These uncertainties are incorporated into their mathematical TP model. Other works also identify the different sources of uncertainties for their probabilistic TP models (Swierstra and Green, 2003; Mueller et al., 2002). Another probabilistic model is proposed for predicting position in the near-term and mid-term future (Prandini et al., 2000). They derive an expression for the probability of conflict using the provided quantitative bounds. Based on the expression, the model is able to generate potential fields for probabilistic path planning of the aircraft. MC simulations is used to validate the purposed method. Their following work builds a probabilistic aircraft position forecasting model using the flight plan and current position (Lygeros and Prandini, 2002). Aircraft's intent is also an important factor in probabilistic TP models. Researchers have incorporated aircraft's intent into flight dynamics to predict future positions as a series of probability density functions (Liu and Hwang, 2011).

Weather Impact on Conventional TP: Several existing studies focus on the impact of convective weather to the special circumstances of a TP task (Mondoloni, 1998; Ng et al., 2009; Stewart et al., 2012, 2013). Genetic algorithm (GA) is used to provide fuel and time-saving flight plans (Mondoloni, 1998). This method takes tailwind into account for the best route to meet specific

requirements. Dynamic programming has been used to search for the best fuel and time-saving model, utilizing the convective weather avoidance model (Ng et al., 2009). The development of dynamic weather reroutes (DWR) (McNally et al., 2012; Erzberger et al., 2010, 2012) is another approach to studying the weather impact on TP. It is proposed as a ground-based concept to automatically and efficiently propose short-time trajectories avoiding convective weather regions for in-flight aircraft in the en-route airspace (McNally et al., 2012). Based on this concept, the system Terminal AutoResolver was deployed to find the auxiliary waypoint of the optimal path around the convective weather polygon (Erzberger et al., 2012). A tactical rerouting model around convective weather was proposed in Stewart et al. (2012), and their later work found flight information further improves the prediction results (Stewart et al., 2013).

2.2.2. Machine learning methods

The ML-based approach is also widely used for TP. Due to the nature of ML techniques, most of the work utilized a massive amount of real history recorded data for the model training. A model using GLMs to perform TP is proposed in De Leege et al. (2013). The model is trained with labeled inputs such as aircraft type, ground speed, altitude, surface wind, etc. The prediction is made through a step-wise regression. Recent work on dropout-based BDL is proposed for TP-based safety assessment in Zhang and Mahadevan (2020), which adopts dropout-based BDL to predict the trajectory deviations. Based on the predictions, this paper proposes a method for aircraft safety assessment.

Weather Impact on Machine Learning Methods: Weather information is included and modeled as grids or cubes in several research papers (Ayhan and Samet, 2016; Liu and Hansen, 2018; Pang and Liu, 2020a,b). The airspace is considered as a 3D grid network and the weather information is stored in each grid in Ayhan and Samet (2016). The raw trajectories are aligned with weather information and made these grid 4D joint cubes, which contains the latitude, longitude, altitude, and the weather condition (representing the time dimension). Then an HMM model is used to predict trajectories with weather uncertainties. Another study using a deep generative convolutional recurrent neural network approach for 4D trajectory prediction is the first paper using an encoder–decoder recurrent neural structure for TP (Liu and Hansen, 2018). The paper proposes an end-to-end convolutional recurrent neural network that consists of a long short-term memory (LSTM) encoder network and a mixture density LSTM decoder network. The model can predict the aircraft 4D trajectories using high-dimensional weather features and last field flight plans. The prediction error metrics show that average absolute horizontal errors are around 50 nautical miles and 2800 ft for average vertical errors. Our previous work on TP follows this direction, in which a conditional generative adversarial network (CGAN) is proposed for TP within each TRACON (Pang and Liu, 2020a) using the data from the Sherlock Data Warehouse (SDW). The convective weather data are stored in a high-dimensional tensor. The drawbacks of CGAN are training difficulties and the source of uncertainty comes from the sampled input. Another previous work also adopts the dropout-based BDL for probabilistic trajectory prediction and shows effectiveness (Pang and Liu, 2020b). However, the dropout ratio is defined manually during the training and the predicted TP uncertainties are strongly correlated with this model parameter. No theoretical method is available to determine the dropout ratio and the trial-and-error is performed. Thus, the uncertainty associated with this subjective ratio is inevitable. A more reliable and robust approximation for the BDL model is needed.

3. Bayesian deep learning

In this section, we will give a brief review of the concept and mechanism of NN in Section 3.1. Following this, we will describe the formulation of approximate variational inference (VI) to solve Bayesian neural network, which treats NN weights as probabilistic distributions in Section 3.2.

3.1. Deterministic TP: Deep learning

The deep learning model is defined as the stacking of many NN layers as a deep model to perform supervised learning, semi-supervise learning, or unsupervised learning task. Different types of deep learning models may consist of different types of layers. Feedforward neural network consists of multiple fully-connected layers. Recurrent neural network (RNN) consists a stack of recurrent layers (long short-term memory (LSTM) Hochreiter and Schmidhuber, 1997, gated recurrent unit (GRU) Chung et al., 2014) and full-connected layers. Similarly, a convolutional neural network (CNN) is the combination of convolutional layers and fully-connected layers (Krizhevsky et al., 2012). The recent work build a recurrent-convolutional neural network (RCNN) model by integrating deterministic CNN and RNN for spatial–temporal learning (Xingjian et al., 2015).

Fig. 1 shows a simple fully connected neural network (FCNN) architecture. The inputs are X_1 and X_2. The output is Y. The hidden units are H_1, H_2, H_3 and H_4. The arrows represent the parameters or weights of the model. Typical NN parameters include weights, and also bias terms. The fully connected layer is named as such because each output of the fully connected layer is a weighted sum of each input element. In Eq. (1), the input is X and the parameters are the weight W and the bias b. f_{FC} stands for a fully-connect layer in NN.

$$f_{FC}(X) = X * W^T + b \tag{1}$$

The architecture shown in Fig. 1 has three layers. The input layer, the hidden layer, and the output layer. The hidden layer has two input nodes, the hidden layer has three nodes and the output layer has one node. The output of the neural network is passed through an activation function to cap the output into the desired range. The optimization objective is defined as the difference measure between the NN output and the true label data. We minimize the optimization objective to get the best parameters that

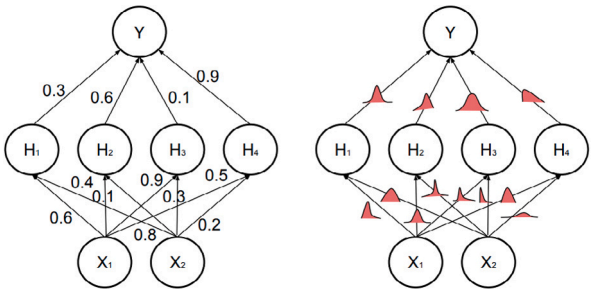


Fig. 1. Left: Classical neural network, each weight has a fixed value. Right: Bayesian neural network, each weight is represented by a distribution.

can generate the closest output to the true labels. This process is also called *training*. The prediction output is made by passing the test data inputs into the NN with the best parameters.

The formulation of CNN follows a similar procedure as feedforward NN but with convolutional operations as shown in Eq. (2). The symbol $\otimes$ stands for convolution operations. In practice, learning the function with fully connected layers may be difficult because of the large number of parameters. For example, if our input is a $224 \times 224 \times 3$ image, and our fully connected layer has a modest output size of 500, then we need $3 \times 224 \times 224 \times 500 \approx 75$ million weights for a single layer. This is not desirable as our dataset may has much fewer samples than this, so the model will suffer from underfitting. One solution to this problem is to utilize weight sharing, of which convolutional neural networks are one popular type.

$$f_{CONV}(X) = X \otimes W^T + b \tag{2}$$

Convolution operations are useful to model the spatial data, when we expect our data to exhibit translation invariance in some dimensions, such as the image data. For example, if we are trying to detect a car, it would be useful to detect wheels. These wheels could be anywhere in the image, so we would like our wheel detector to be translationally invariant. As in the fully connected networks, a stacked network of convolutional layers can learn complex functions of the input data.

The convolution operation uses a kernel, or filter, to compute outputs. In convolutional networks for image recognition, the filter size is usually small (3×3 or 5×5 for 2D convolution). The weights of the filters are learnable parameters. Additionally, the convolutional layer usually incorporates a learnable bias term, which is added to the filter output. As the filter slides across the input, outputs are generated for each input location. At the borders of the image, input data for the convolution is missing. This data can be replaced with zeros (zero paddings), or the convolution output at these locations can be ignored (valid paddings). RNN is relatively a complicated flow of computation. LSTM is a special form of RNN and achieves great success in the learning of sequence data. GRU is viewed as a simplified version of LSTM. All recurrent neural networks have a repeating module of the neural network called the recurrence. We list the calculation of one LSTM recurrence here.

$$f_t = \sigma(W_f \cdot h_x + b_f) \tag{3}$$

$$i_t = \sigma(W_i \cdot h_x + b_i) \tag{4}$$

$$\hat{c}_t = tanh(W_c \cdot h_x + b_c) \tag{5}$$

$$c_t = f_t \cdot c_t + i_t \cdot \hat{c}_t \tag{6}$$

$$o_t = \sigma(W_o \cdot h_x + b_o) \tag{7}$$

$$h_t = o_t \cdot tanh(c_t) \tag{8}$$

The LSTM cell consists of three cell gates, the forget gate in Eq. (3), the input gate in Eq. (4), and the output gate Eq. (7). These three gates are stringed together by the cell state tensor as Eqs. (5) and (6). The forget gate is to decide whether the value needs to be passed through the cell state or not. The input gate is to control if the value needs to be stored in the cell state. The output of the model is decided by the calculation in Eqs. (7) and (8). We should notice the usage of the activation functions σ and $tanh$ here. The σ and $tanh$ push the output into the range (0, 1) and (−1, 1), respectively. One of the key ideas behind LSTM is to connect the previous state to the current state, which is useful to model the complex sequential dependency. This is accomplished by the hidden tensor and the cell tensor through the calculation of three different gates.

To avoid overfitting, various stochastic regularization techniques (SRTs) are developed such as the use of early stop, data augmentation, parameter penalties, dropout, and Bayesian regularization. The network with Bayesian regularization is prone to resist overtraining and overfitting because the prior information provides a natural tendency to select simpler models. Furthermore, the Bayesian criterion provides a natural way of stopping training and tuning parameter selection (Burden and Winkler, 2008).

3.2. Probabilistic TP: Bayesian deep learning

The formulation of Bayesian Deep Learning relies on Bayesian probabilistic modeling. The simple idea is to have stochastic NN parameters, as the right part in Fig. 1. For a regression task given a training input sequence $X = \{x_1, \ldots, x_n\}$ and their corresponding

Table 1
Notations in BDL.

Notations	Meaning
X	The training inputs.
Y	The prediction outputs.
ω	The model parameters.
x^*	The new observed inputs.
y^*	The predictions corresponding to x^*.
q_θ	The approximated variational distribution

output sequence $Y = \{y_1, \dots, y_n\}$, we try to find the parameters ω for the approximation function $y = f^\omega(x)$ that are *most likely* to generate the outputs. That is, the inference of $p(\omega|X,Y)$ (see Table 1 for the notations in BDL).

$$\overbrace{p(\omega|X,Y)}^{\text{Posterior}} = \frac{\overbrace{p(Y|X,\omega)}^{\text{Likelihood}}\overbrace{p(\omega)}^{\text{Prior}}}{\underbrace{p(Y|X)}_{\text{Evidence}}} \tag{9}$$

The Bayesian approach gives a space of parameters ω as a distribution $p(\omega)$ called the *prior*. The *prior* is defined based on prior knowledge and gives us an adversarial of what parameters are likely to generate our data before we have any data points. The Bayesian approach also defines a likelihood distribution $p(y|x,\omega)$, which is a probabilistic model of the model outputs given the data points and model parameters. The normalizer in Eq. (9), $p(Y|X)$, is the evidence (Gal, 2016). The posterior $p(\omega|X,Y)$ is evaluated with Eq. (9).

For the prediction of the model, if we have a new observed data input sequence x^*, the predicted output sequence y^* is a distribution, marginalized over the posterior, as in Eq. (10). We can also view it as a weighted average of the model where the weights are determined by the posterior distribution of ω, mathematically, $\mathbb{E}_{p(\omega|X,Y)}[p(y^*|x^*,\omega)]$. This is also equivalent to using an ensemble of NNs for prediction. Unfortunately, this is intractable for any practical case (Blundell et al., 2015).

$$p(y^*|x^*,X,Y) = \int p(y^*|x^*,\omega)p(\omega|X,Y)d\omega \tag{10}$$

The normal way to define the prior distributions is to place a standard Gaussian distribution over each of the NN weights W while the bias b is a deterministic value (Neal, 2012). Due to the non-conjugacy and non-linearity of the complex structure of NNs, the closed-form derivation for Bayesian inference to posterior is intractable. On the other hand, the traditional sampling-based methods lead to surprising prohibitive computational complexity, and lack of scalability to large scale practical application problems.

Variational inference, as an approximated posterior inference algorithm, has been applied to the posterior inference of Bayesian neural networks. First, we define an approximated variational distribution $q_\theta(\omega)$ for the posterior $p(\omega|X,Y)$. Kullback–Leibler (KL) divergence (Kullback and Leibler, 1951) is adopted here to measure the difference between these two distributions. Thus minimizing the KL divergence w.r.t. θ leads to the best approximated variational distribution.

$$\begin{aligned} &KL(q_\theta(\omega) \parallel p(\omega|X,Y)) \\ &= \int q_\theta(\omega) log \frac{q_\theta(\omega)}{p(\omega|X,Y)} d\omega \\ &= \int q_\theta(\omega) log \frac{q_\theta(\omega)p(X,Y)}{p(\omega,X,Y)} d\omega \\ &= \underbrace{\log p(X,Y)}_{\text{Constant}} - \underbrace{\int q_\theta(\omega) log \frac{p(\omega,X,Y)}{q_\theta(\omega)} d\omega}_{\text{Evidence Lower Bound (ELBO)}} \end{aligned} \tag{11}$$

After rearranging Eq. (11), the minimization of KL divergence is equivalent to minimize the negative Evidence Lower BOund (ELBO).

$$\begin{aligned} -ELBO &:= KL(q_\theta(\omega) \parallel p(\omega)) - \int q_\theta(\omega) log(p(Y|X,\omega)) d\omega \\ &= \underbrace{KL(q_\theta(\omega) \parallel p(\omega))}_{\text{Prior dependent}} - \underbrace{\mathbb{E}_{q_\theta(\omega)}[log(p(X,Y|\omega))]}_{\text{Data dependent}} \end{aligned} \tag{12}$$

The negative ELBO is a sum of a prior dependent part and a data-dependent part (Eq. (12)). The prior dependent part can be referred to as complexity cost and the data-dependent part can be viewed as the likelihood cost. The negative ELBO embodies a trade-off between meeting the complexity of the dataset (X,Y), and satisfying the simplicity of prior (Blundell et al., 2015). Again, the minimizing of negative ELBO is intractable but luckily approximations and gradient-based methods can help us to find the optimum during inference.

MC integration is introduced to approximate the data-dependent part. The basic concept of MC integration is to substitute integrals with summations. The mean-field assumption of variational inference is used in MC integration, for the sake of simplicity. The mean-field variational family (Blei et al., 2017) assumes each of the latent variables is mutually independent and each is governed by a distinct factor. In Eq. (13), each of the latent variable ω_i is governed by its own variational factor, the density $q_{\theta_i}(\omega_i)$.

$$q_\theta(\omega) = \prod_{i=1}^{m} q_{\theta_i}(\omega_i) \tag{13}$$

Under this assumption, the data-dependent part can be evaluated in a relevant simple manner. Specifically, we sample $\hat{\omega}$ from the variational distribution $q_\theta(\omega)$ and optimize towards the objective function w.r.t. θ in Eq. (14) each time. By doing this recursively, we find the best $q_\theta(\omega)$ that approximated the true posterior. The final objective, or loss function for NN, is defined as the summation of KL divergence between the prior and the approximated variational distribution, and the negative log-likelihood (NLL). The method that samples the weight of a neural network stochastically at training time is referred to as the weight perturbation method (Wen et al., 2018).

$$-\widehat{ELBO} := \underbrace{KL(q_\theta(\omega) \parallel p(\omega))}_{\text{KL divergence}} \quad \underbrace{-log(p(X,Y|\omega))}_{\text{Negative Log-Likelihood (NLL)}} \tag{14}$$

Once the training process is finished, the uncertainty can be estimated through MC tests as in Eq. (15). K is the number of tests performed.

$$\begin{aligned} p(y^*|x^*, X, Y) &= \mathbb{E}_{p(\omega|X,Y)}[p(y^*|x^*, \omega)] \\ &\approx \frac{1}{K}\sum_{k=1}^{K} p(y^*|x^*, \hat{\omega_k}) \end{aligned} \tag{15}$$

Gal and Ghahramani showed that a neural network with dropout applied before every layer is mathematically equivalent to the Bernoulli approximate VI of NNs (Gal and Ghahramani, 2016a). The following work of them shows similar behavior to convolutional layers (Gal and Ghahramani, 2015), and recurrent layers with variational dropout applied (Gal and Ghahramani, 2016b). The dropout approximation to Bayesian NNs is also known as the Monte Carlo Dropout (MCDropout) method. This work draws large attention because only minimal change is needed to switch the current deep learning models into Bayesian deep learning models. The uncertainty is estimated by simply applying dropout during the testing procedure. Consequently, abundant applications of dropout approximation to a Bayesian posterior raised, as well as in the field of ATM (Zhang and Mahadevan, 2020; Pang and Liu, 2020b). However, researchers have queried MCDropout (Pearce et al., 2018; Hron et al., 2018; Osband et al., 2018; Melis et al., 2018). A few major concerns are,

- The improper prior in variational dropout leads to irremediably pathological behavior of the true posterior.
- The dropout ratio defined manually leads to arbitrarily poor decision making.
- The unsatisfactory on lack of theoretical grounding for dropout, without which the choice of dropout variant remains arbitrary.

A reliable uncertainty estimate is critical for enhancing the safety in aviation operations, such as uncertainty quantification of TP for early trajectory conflict awareness. It is better to constraint the KL divergence explicitly in the objective functions, as in Eq. (14). Weight perturbation methods such as the local reparameterization trick (LRT) (Kingma et al., 2015) and Flipout (Wen et al., 2018) has enabled the possibility of optimizing towards Eq. (14) directly using sampling and gradient-based optimization during inference. The recent development of probabilistic programming languages build the pathway of fast and scalable application of neural network uncertainty estimations when dealing with real-world big data (Dillon et al., 2017; Tran et al., 2016, 2018, 2019).

4. Proposed methodology for TP under uncertainties

The problem definition and overview of the proposed method will be discussed in Section 4.1. Fig. 2 visualizes the innovative information fusion pipeline of the proposed problem-solving framework. The architecture can be divided into three modules, the data processing and filtering module, the feature extraction module, and the Sequence to Sequence (Seq2Seq) learning module. The data processing module is introduced in Section 4.2. Then the weather feature extraction module and the Seq2Seq learning module are described in Section 4.3. Finally, the implementation and model inference details are discussed in Section 4.4.

4.1. Overview

4.1.1. Problem definition

First, we would like to introduce several abbreviations involved, as shown in Table 2.

The following data is used in our problem:

- **Trajectory data** Y^{true}, which is the actual trajectory of the aircraft, with dimension $[N \times L \times W]$.
- **Flight plan data** X^{plan}, which refers to the processed last on-file flight plan of the aircraft prior to takeoff, with dimension $[N \times L \times W]$.

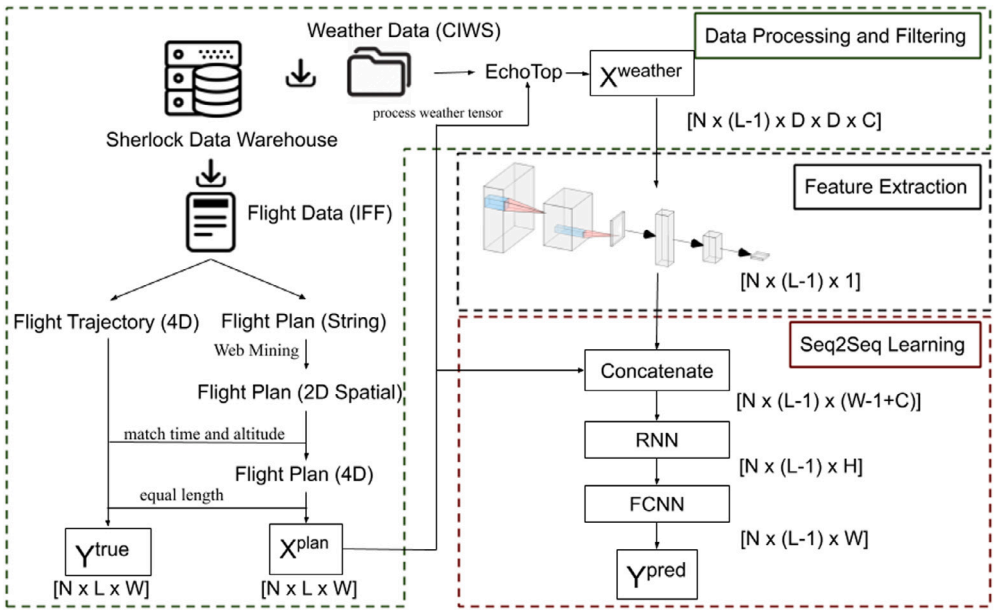


Fig. 2. Overview of the proposed trajectory prediction framework. The ML pipeline consists of the data processing and filtering module, feature extraction module, and Seq2Seq learning module.

Table 2
Abbreviations.

Notations	Meaning
N	Number of Samples/The number of flight trajectories in each sector. The total number of the data is separated for model training and testing. N is different for different sections.
L	Length of sequence. L represents the number of aircraft positions in one recorded trajectory, corresponding to one flight call sign. L is set to be 50 in our case, for the simplicity of the demonstration.
W	The dimension the prediction of the flight trajectories. W is 3 in our case, which represents the latitude, longitude, and altitude dimension.
D	Spatial resolution of the weather window. In our case, D is set to be 32, which implies that the weather window is 32×32. The selected window size is based on the approximated distance to cover the look-ahead region at the current coordinates.
C	Channel number of weather features. C is 1 in our experiments, which only includes the convective weather EchoTop values. Wind, pressure, humidity, etc, can be added to expand the channel number.
H	Hidden dimension of recurrent cell. The number of recurrence in the recurrent layer. A parameter to be determined during the parameter tuning process. 64 is determined in our case.

- **Weather Data** $X^{weather}$, which refers to the processed look ahead convective weather data around each position of the y^{plan}. The dimension of cube is $[N \times L \times D \times D \times C]$.

A more detailed description of the used data is given in Section 4.2. Finally, the following output sequence will be given from the proposed neural network model:

- **The predicted trajectory** Y^{pred}, with the dimension $[N \times (L-1) \times W]$.

The model aims to perform strategic TP based on the last on-file flight plan and weather forecast prior to takeoff. According to this, the TP task is defined as predicting Y^{pred} based on the last on-file flight plan X^{plan} and weather conditions $X^{weather}$ along the planned route X^{plan}. Thus, the model inputs are $X^{weather}$ and X^{plan}, the model output is Y^{pred}. The objective is to minimize the distance between Y^{pred} and Y^{true}.

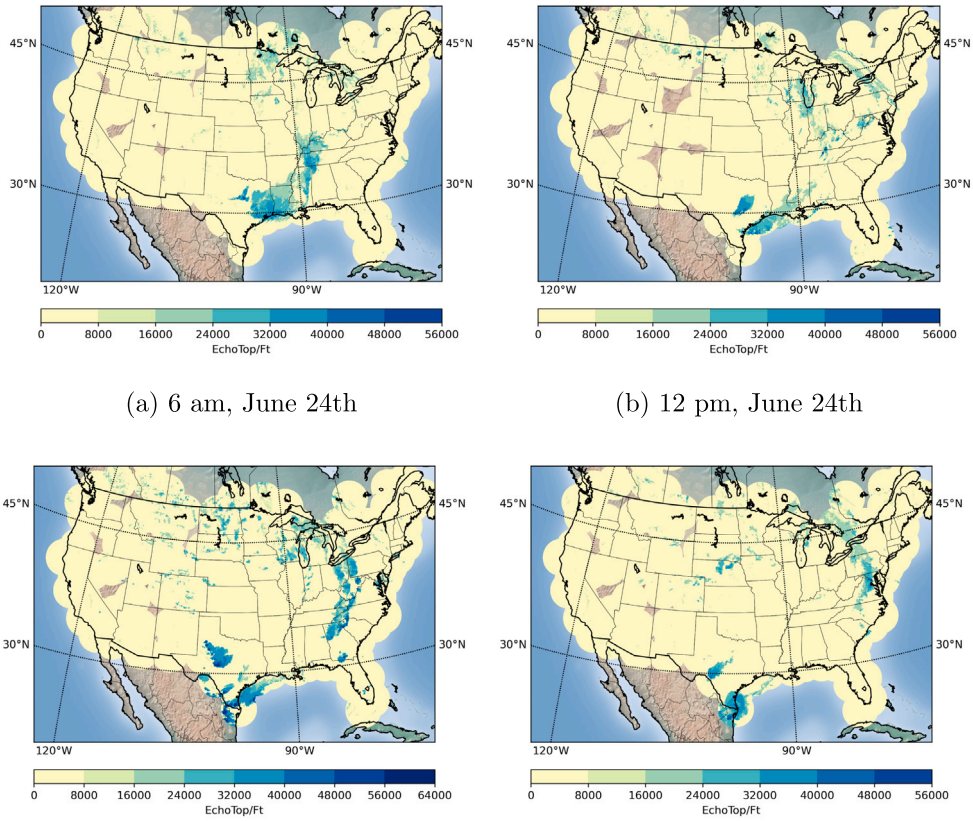


Fig. 3. EchoTop convective weather visualization on June 24th, 2019 at (a) 6 am; (b) 12 pm; (c) 6 pm; (d) 12 am.

Both Deterministic TP and probabilistic TP models are implemented and evaluated under the same problem definition and dataset. Probabilistic TP model includes both the MCDropout approximation of BDL and the VI based BDL. A comparison study is performed on various methods in Sections 5 and 6.

4.2. Data processing and filtering module

The data used in this research is obtained from the Sherlock Data Warehouse (SDW) (Arneson, 2018). It is a platform for reliable aviation data collection, archiving, processing, query, and delivery to support ATM research. Data of Sherlock comes primarily from the federal aviation administration (FAA) and the National Oceanic Atmospheric Administration (NOAA) (Eshow et al., 2014). Multiple sources of raw surveillance data are processed and stored in SDW. Here we only use the data from two sources. The Integrated Flight Format (IFF) flight data and EchoTop (ET) convective weather data from CIWS. To properly address the impact of weather on trajectory prediction, we look into the historical weather recordings for a specific day with convective weather presence. The NOAA's storm prediction center reported convective weather was formulated, with tornado and high winds in the north-east U.S. airspace on June 24th, 2019.

4.2.1. Flight data processing

The flight data is collected from different FAA facilities and stored as IFF format. It includes all raw source data plus the derived fields such as flight summary, track points, and flight plan. The flight summary is a general description of the flight, which contains flight time, flight call sign, aircraft type, origin, and destination information. The flight track points are the record of real flight operation. It includes the ground measured aircraft position in both the spatial and temporal domains. The flight plan comes as a string of waypoints. We developed a web-based mining tool to translate it into WGS84 coordinates. We choose five air traffic control centers for our demonstration since most of the tornado and high winds are reported around this area. Fig. 4 visualizes the flight tracks flying inside these control centers. These flight tracks with reference to the convective weather conditions are shown in Fig. 3.

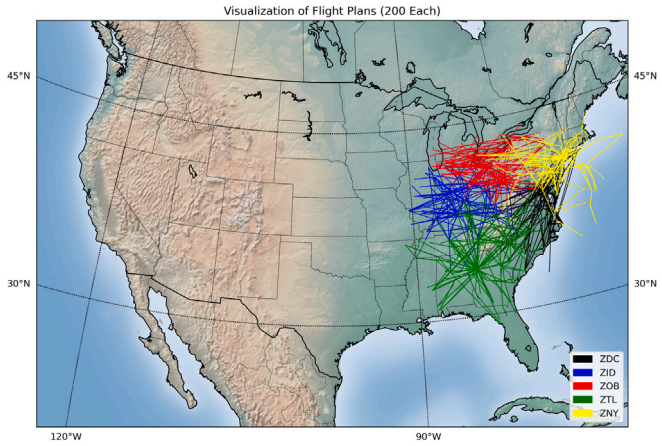


Fig. 4. Visualization of sampled flight plans in different sectors on 06/24/2019.

Table 3
Weather encounters classified for different sectors.

	Total	Weather encounters	Percentage
ZID	4103	2817	68.7%
ZTL	6255	3642	58.2%
ZNY	6135	3757	61.2%
ZDC	5420	2907	53.6%
ZOB	4771	2333	48.9%

Fig. 2 shows the procedure of processing sector flight data. The flight tracks are extracted from the sector IFF file and a linear interpolation is performed for each of the track points with a 1-second interval. The waypoint of each flight plan are removed if the waypoint is outside the investigated flight control centers in this study. The Euclidean distance is used to evaluate whether waypoints with respect to the current sector range. The same interpolation is performed for the flight plan to make it consistent with the track points. Finally, the sequence from the track points and flight plan interpolation are used in the proposed model.

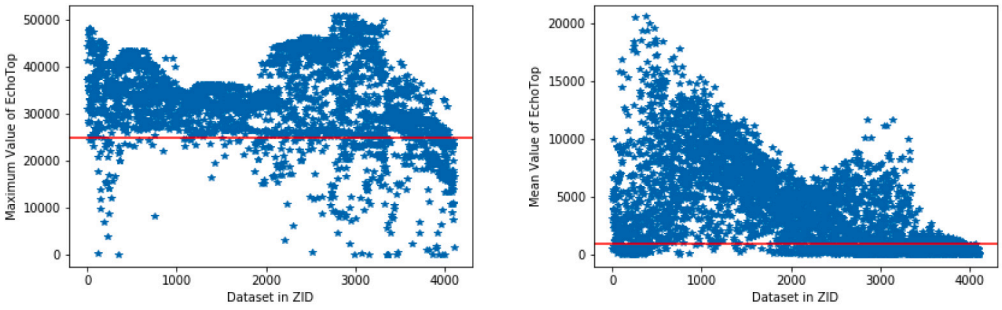
4.2.2. Weather data processing

The weather data are obtained from Corridor Integrated Weather Systems (CIWS) in SDW (Klingle-Wilson and Evans, 2005). CIWS is designed to improve convective weather decision support for the congested en-route airspace. The two key features of CIWS, i.e., EchoTop (ET) and Vertically Integrated Liquid (VIL), come with current and forecast dataset in the Sherlock database. EchoTop numbers are estimates in feet of the highest cloud tops associated with the radar echoes. The higher cloud tops usually indicates the more intense precipitation. VIL is an estimate of the total mass of precipitation in the clouds. The measurement is obtained by observing the reflectivity of the air which is obtained with weather radar. Reflectivity represents the intensity of radar echoes returning from clouds. ET and VIL are updated and documented every 150 s, which results in 576 files daily.

In this research, we only use the current ET data for demonstration. At each given flight plan points in Y^{plan}, a weather cube ahead of the current location is selected from the original weather file. The cube direction is rotated to align with the heading angle of the aircraft. This is the same weather cube generator used in our previous work (Pang et al., 2019). The cube size is selected to be 32×32. The reason for this selection is because the size will cover the look-ahead region for decision making of possible weather avoidance. Thus the final dimension of the weather feature cubes is $N \times 49 \times 32 \times 32 \times 1$ and the dimension of the flight tracks and flight plans are $N \times 49 \times 3$, where N is the number of data processed for each sector. It should be noted that the weather data of the starting point in Y^{plan} is not processed, as it is the current location of the aircraft.

4.2.3. Data filtering

Past study on the convective weather dataset indicates that only the flight tracks that have encountered convective weather ET of 25,000 ft or continued for at least 2 min are classified as weather encountered deviations (DeLaura et al., 2008). In considering this, the flight track and convective weather dataset are filtered by the maximum and mean value of ET of each data pair, as shown in Table 3. The threshold value for maximum value ET and mean value ET are 25,000 and 1000, respectively. A visualization of the data screening process is shown in Fig. 5. The x-axis stands for the 4103 pairs of data for sector ZID and the y-axis represents the maximum and mean values of each data pair. The red line visualizes the threshold values used in different metrics.



(a) Maximum EchoTop (b) Mean EchoTop

Fig. 5. Data filtering with threshold on EchoTop. Visualizing of ZID.

Table 4
Architecture of feature extraction module and Seq2Seq learning module with processed recorded dataset.

Layers	Input size	Output size	Parameters
Conv3d_1	[N, 49, 32, 32, 1]	[N, 49, 10, 10, 3]	Kernel: [1, 5, 5, 3], Stride: [1, 3, 3, 1], No Padding
Conv3d_2	[N, 49, 10, 10, 3]	[N, 49, 4, 4, 1]	Kernel: [1, 3, 3, 1], Stride: [1, 3, 3, 1], Zero Padding
Flatten	[N, 49, 4, 4, 1]	[N, 49, 16]	Keep first two dimension
Dense_1	[N, 49, 16]	[N, 49, 4]	4
Dense_2	[N, 49, 4]	[N, 49, 1]	1
Concat	[N, 49, 1]	[N, 49, 4]	Concatenate with flight plan p
LSTM	[N, 49, 4]	[N, 49, 64]	64
Dense_3	[N, 49, 64]	[N, 49, 32]	32
Dense_4	[N, 49, 32]	[N, 49, 16]	16
Dense_5	[N, 49, 16]	[N, 49, 3]	3

4.3. Feature extraction module and Seq2Seq learning module

The feature extraction module and Seq2Seq learning module can perform deterministic TP with classical deep learning layers as introduced in Section 3.1. They can also be easily changed into probabilistic TP when the Bayesian theory is integrated with Deep Learning as shown in Section 3.2. Feature extraction is performed to extract the indication of weather conditions at each point within the entire sequence. The extracted weather feature is combined with the flight plan as the input to the Seq2Seq learning module. The feature extraction module and Seq2Seq learning module form the complex spatial and temporal structures of the weather information and flight trajectory. These two models can be made deterministic for the point estimate of flight trajectories or probabilistic to quantify the flight trajectory uncertainty.

The weather feature extraction module includes two three-dimensional convolutional layers and two fully-connect layers. The convolutional layers are used to extract useful features from $cube$ into a vector. The extracted weather vector will be concatenated with the flight plan tensor y^{plan} as the inputs to the Seq2Seq learning module. The Seq2Seq learning module consists of one recurrent layer and three fully-connect layers, which has been shown to have a supreme performance on sequence learning tasks in the literature (Sutskever et al., 2014). The concatenation layer connects the feature extraction module and the Seq2Seq learning module. The prediction y^{pred} is the output from the last fully-connect layer. A detailed introduction of these two modules is also shown in Table 4. In this table, N represents the batch size and empty cell means not applicable. The layer names that use bold font means need an input.

The first input to the model is the weather cube tensor $cube$ with dimension $[N, 49, 32, 32, 1]$. The Conv3d_1 layer has a kernel size of $[1, 5, 5, 3]$ and a stride of $[1, 3, 3, 1]$ along the last four dimensions of the input tensor with no padding added for convolution operation. The output has a dimension of $[N, 49, 10, 10, 3]$. The second convolutional layer Conv3d_2 has a similar setup as the first one with a kernel size of $[1, 3, 3, 1]$, a stride of $[1, 3, 3, 1]$, and zero padding, respectively. The output has a size of $[N, 49, 16]$ after flattening along the last three dimensions of the tensor. The following two dense layers would compress the tensor to a dimension of $[N, 49, 1]$, which is the dimension of the weather feature tensor. We choose to concatenate it with the flight plan tensor p as the input to the RNN cell. The dimension of LSTM hidden state h_t and cell state c_t tensor is 64. Zero initialization is used in the LSTM cell. The output of the LSTM cell will be fed into three dense layers to calculate the output of the model. The training objective is defined based on Eq. (14) which incorporates two parts: the KL divergence and the NLL.

4.4. Integration and inference

4.4.1. Metrics for performance evaluation

For the deterministic TP model, the training objective is defined as the minimization of the difference between the predicted sequence Y^{pred} and the ground truth sequence Y^{true}, as in Eq. (16). It is defined as the root mean squared error (RMSE) between

the two sequences. K is the size of the training dataset.

$$RMSE = \sqrt{\sum_{k=1}^{K} \frac{(Y_k^{pred} - Y_k^{true})^2}{K}} \tag{16}$$

For the probabilistic TP model, the training objective is derived in Eq. (14). This can be reformulated to accommodate our proposed framework as in Eq. (17). The KL-divergence of the variational posterior $q_\theta(\omega)$ and prior $p(\omega)$ for each of the parameter ω can be calculated analytically if a normal distribution is assumed for both of them. The NLL term is the RMSE loss defined above. Once the inferences of network parameters are finished, the mean predictions and corresponding uncertainties can be evaluated by MC tests, as in Eq. (15).

$$-ELBO = KL(q_\theta(\omega), p(\omega)) + RMSE \tag{17}$$

4.4.2. Implementation details

The entire dataset is separated into the training set and the testing set with a ratio of 0.8 and 0.2. The training set is used in the model training process solely and the testing set is for model testing after the training procedure. Furthermore, the validation set is sampled from the training set to evaluate the model performance during the training process. Data normalization and denormalization are performed prior to and after the model training.

Adam optimizer (Kingma and Ba, 2014) is used to minimize the objective until the negative ELBO converges. The initial learning rate is set to be $1e^{-3}$ by *best practice* during fine-tuning of the model parameters. Learning rate decay follows Eq. (18),

$$lr^{s+1} = \frac{lr^s}{(1 + r * s)^q} \tag{18}$$

where s is the global step during optimization, the decay rate r is $1e^{-4}$, and the power q equals to 0.75.

The training is performed with the deep probabilistic learning packages Edward (Tran et al., 2019, 2016, 2018) and Tensorflow-probability (Dillon et al., 2017) based on Tensorflow 2.3.0 on a workstation with Intel Xeon E5-1620 v4 @3.50 GHz CPU and accelerated with a single NVIDIA GTX 1080 GPU.

The comparison with the MCDropout method and deterministic model is conducted with the same model parameters and training setups. The use of dropout operation can be summarized as, (a) applying regular dropout after each convolutional and dense layers; (b) performing variational dropout (Gal and Ghahramani, 2016b) to LSTM cell. It should be noted that there is no dropout operation after the last dense layer. The dropout ratio used in this paper is set to 0.9.

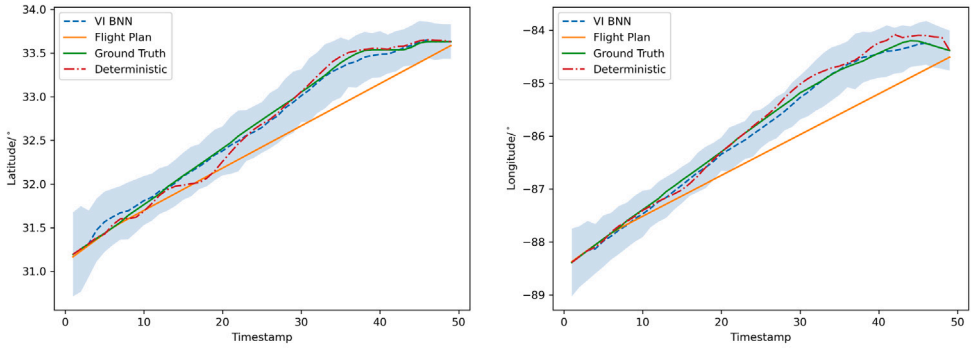
5. Demonstrations and results

In this section, we first evaluate the prediction of the TP framework individually with the test data in Section 5.1. This includes the discussions on several good prediction results and different reasons for failed cases. Next, a statistical study for the entire test dataset is performed to evaluate the overall TP uncertainty reduction performance in Section 5.2. The comparison of all methods mentioned in the above sections is also performed.

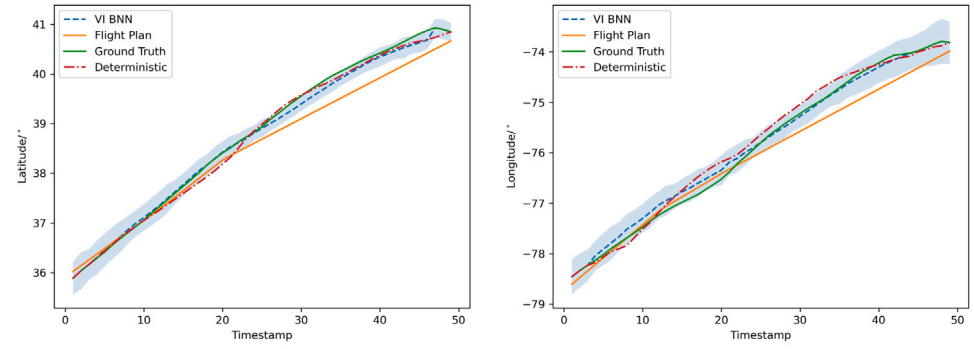
5.1. Individual evaluation

Figs. 6 and 7 visualize model predictions from the test dataset. In order to better represent the 95% confidence interval on both latitude and longitude dimension, we plot the latitude and longitude separately. The x-axis is the timestamp for the entire trajectory sequence. As the inputs to the TP framework, it is shown that the actual flight trajectories Y^{true} (ground truth) deviate from the last on-file flight plans Y^{plan}. On the other hand, our model prediction Y^{pred} discovered the deviation from Y^{plan} and calibrated the outputs towards Y^{true}. The two cases in Fig. 6 reveal the effectiveness of the proposed ML pipeline. In case 1, both the deterministic prediction and the VI BNN prediction are closer to the ground truth compared with the flight plan. The 95% confidence interval acts as a reliable safety assurance on the predicted trajectories. The mean prediction of VI BNN shows smaller oscillations which comes from the intrinsic Bayesian regularization of NNs. Case 2 is another test example from the test dataset of ZDC. In this case, we noticed that the longitude shows a higher accuracy compared to the latitude, due to the dynamic formulation of the aircraft motion. This can be further investigated when the aircraft motion recording data (e.g. gyroscope measurements) is available.

On the other hand, Fig. 7 shows two typical failed cases of the proposed ML pipeline. In case 3, the aircraft ignored the assigned waypoint at timestamp 34. As the input of the ML pipeline, the flight plan has a large impact on the outputs. Although the overall trajectory deviation is reduced compared with the flight plan, the figures show unsatisfied predictions. In this case, the 95% confidence interval prediction based on VI BNN is able to cover the ground truth most of the time. Case 4 shows an extreme case where a large deviation and sharp turn in the ground truth presented due to severe weather conditions. This is one of the drawback of data-driven TP models (Yu et al., 2020). In this case, the deterministic method fails for trajectory deviation reduction. The VI BNN method exhibits benefits with the help of the uncertainty quantification (UQ) capabilities. More accurate prediction for these cases will need other information, such as voice communication data between pilot and controller, to fully explain the discrepancy.



(a) Case 1: Both deterministic and probabilistic predictions show good prediction capabilities. The VI BNN prediction has smaller oscillations compared to deterministic prediction due to Bayesian regularization.



(b) Case 2: Both deterministic and probabilistic predictions show good prediction capabilities. The longitude shows a considerable accuracy compared to the latitude, potentially due to the dynamic formulation of the aircraft motion.

Fig. 6. The model performs good by making predictions closer to the ground truth. The 95% confidence bound for VI BNN is colored in blue. The start point (timestamp 0) is not visualized.

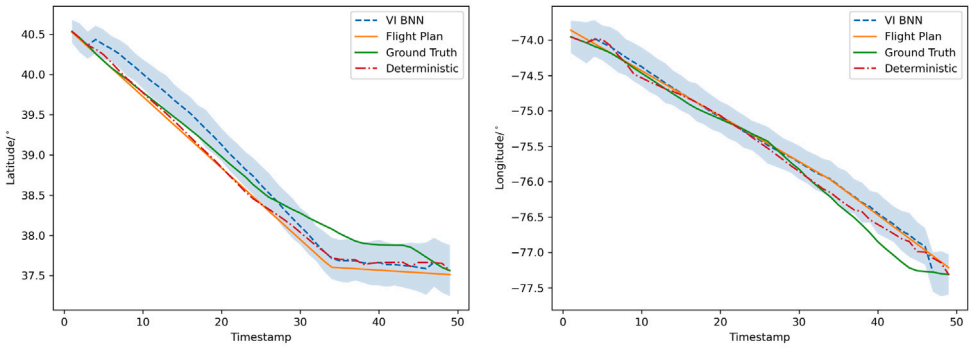
5.2. Statistical evaluation

In this section, we evaluate the effectiveness of the proposed model statistically on the entire test dataset. The key metric used is the percentage of the deviation variance reduction for the mean prediction. We compare the deterministic prediction, and the mean prediction of MCDropout and VI BNN. We also study the contribution of feature extraction module towards deviation reduction in this section.

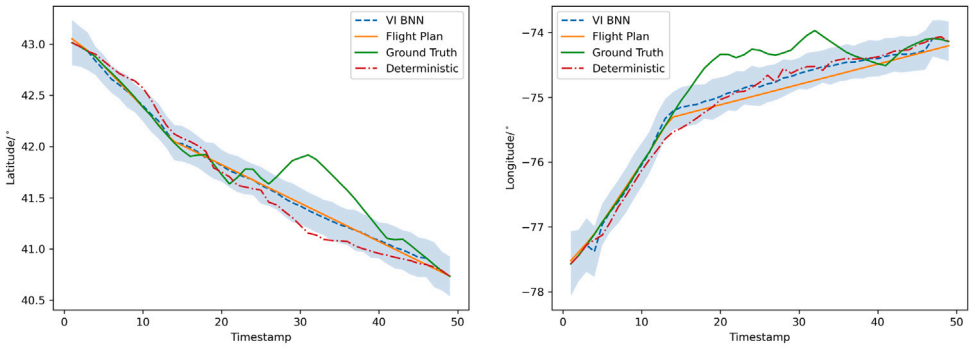
5.2.1. Metrics

The equations used for statistical evaluations are shown in Eqs. (19) and (20). We denote the original deviations as l_{2k}^{ori}, defined as the L_2 distance of the true trajectory y_k^{true} and the flight plan x_k^{plan}. Similarly, the predicted deviation can be defined as sum of squared error of the true trajectory y_k^{true} and predicted trajectory y_k^{pred} as l_{2k}^{new}. The index k is the index of the test dataset. In Eq. (21), we denote ρ_k as the percentage of deviation variance reduction of the prediction l_{2k}^{new} compared to the original variation l_{2k}^{ori} in the kth test data. The subscripts k, n, and d represent the dimensions of the tensors corresponding to the size of the data, length of the sequence, and DOF for the prediction, respectively. In this evaluation, n is 49, and d is 2 for latitude and longitude dimensions. Eq. (21) is the equation we used to calculate the overall variance reduction among the data. The size of ρ is equal to the number of data points with deviation variance reduction.

$$l_{2k}^{ori} = \sum_{i}^{n}\sum_{j}^{d}(y_{k,i,j}^{true} - y_{k,i,j}^{plan})^2 \tag{19}$$



(a) Case 3: The flight plan has a waypoint at timestamp 34 ignored by the ground truth. This waypoint has a large impact on model predictions. However, the 95% confidence interval of VI BNN covers most of the ground truth.



(b) Case 4: The ground truth has a large deviation due to severe weather conditions (≈ 1° latitude/longitude) with sharp turns starting from timestamp 14. The deterministic prediction fails at this case while the VI BNN still exhibits benefits from the confidence interval coverage.

Fig. 7. The model fails at making a good prediction due to various reasons. The 95% confidence bound for VI BNN is colored in blue. The start point (timestamp 0) is not plotted.

$$l_{2_k}^{new} = \sum_{i}^{n}\sum_{j}^{d}(y_{k,i,j}^{true} - y_{k,i,j}^{pred})^2 \tag{20}$$

$$\rho_k = \frac{var(l_2^{ori})_k - var(l_2^{new})_k}{var(l_2^{ori})_k} \tag{21}$$

5.2.2. Deviation reduction and weather effects

In Table 5, we show the percentage of samples that can achieve deviation variation reduction (i.e., $\rho_k < 1$) based on the evaluation metrics described. We also examine the contributions of convective weather. This is achieved by omitting the weather feature extraction module in the ML pipeline. For example, using the deterministic model with weather feature extraction module, 78.5% of the flight trajectory deviations are reduced. The total variance reduction is 22.4% for the test dataset of ZTL. While only 62.3% of the flight trajectory deviations are reduced by 14.5% without the weather data.

We discover that the proposed ML pipeline is effective and shows benefits in all cases, but with different performance for different experiment settings. We also notice that ZTL and ZID has higher percentage of deviation reductions than the other sectors, especially weather-related reductions. One of the reason is that they have higher ratio of weather encounters in flight trajectories. This is verified by checking the movement of the meteorological weather contours in Fig. 3 and Table 3. By feeding the weather features into the ML pipeline, we achieve a higher percentage of flight deviation reduction and a better total variance reduction. Thus, adding

Table 5
Deviation reduction for deterministic (NN) and probabilistic methods (BNN). The probabilistic methods including MCDropout and VI BNN. For probabilistic methods, the results are evaluated with the mean prediction. *Note: In A | B, A is the result with weather features, B is the result without the weather features.*

Model	Flight control sector	Deviation reduced/%	Variance reduction/%
Deterministic	ZTL	78.5% | 62.3%	22.4% | 14.5%
	ZID	84.6% | 46.6%	30.3% | 25.7%
	ZDC	54.4% | 60.8%	36.5% | 26.5%
	ZNY	59.6% | 67.1%	57.6% | 43.2%
	ZOB	66.7% | 44.1%	48.9% | 27.5%
MCDropout	ZTL	45.8% | 39.1%	8.7% | 7.4%
	ZID	36.1% | 11.5%	11.2% | 2.3%
	ZDC	25.4% | 23.7%	3.5% | 6.7%
	ZNY	28.7% | 29.7%	9.8% | 5.8%
	ZOB	14.6% | 17.6%	5.0% | 7.8%
VI BNN	ZTL	69.8% | 55.9%	24.1% | 16.4%
	ZID	79.0% | 22.5%	36.7% | 6.7%
	ZDC	60.7% | 52.5%	46.4% | 16.2%
	ZNY	40.6% | 39.0%	28.6% | 18.3%
	ZOB	25.0% | 11.6%	31.6% | 14.6%

the weather feature extraction module improves the model TP performance by explicitly considering weather uncertainties. On the other hand, the prediction without the weather data also shows considerable deviation reduction. The reason is that our proposed ML pipeline implicitly learn the flight deviations caused by factors other than weather-related reasons from historical record.

Comparing different methods, we realize that deterministic and VI BNN methods has better performance than MCDropout. There are multiple reasons behind this. It is worth pointing out that the dropout ratio used for MCDropout requires hyper-parameter tuning. The training is unstable and hard to converge if we use a large dropout ratio. On the contrast, the uncertainty will be underestimated if we choose a small dropout ratio. We perform hyper-parameter search over the dropout ratios, from 0.1 to 0.9 with interval 0.1 on a sector and choose the best-performed dropout ratio (0.9) for all cases. Also, the Bernoulli assumption on NN parameters improves the difference between each draws during sampling thus introduces more fluctuations. In Table 5, the deterministic approach performs slightly better than VI BNN in several flight control sectors. However, we only list results calculated using the mean prediction for probabilistic methods. The confidence interval of VI BNN significantly improves the confidence by inferring airspace coverage in a probabilistic sense. The oscillation behavior discussed in Section 5.1 of deterministic method is also non-negligible.

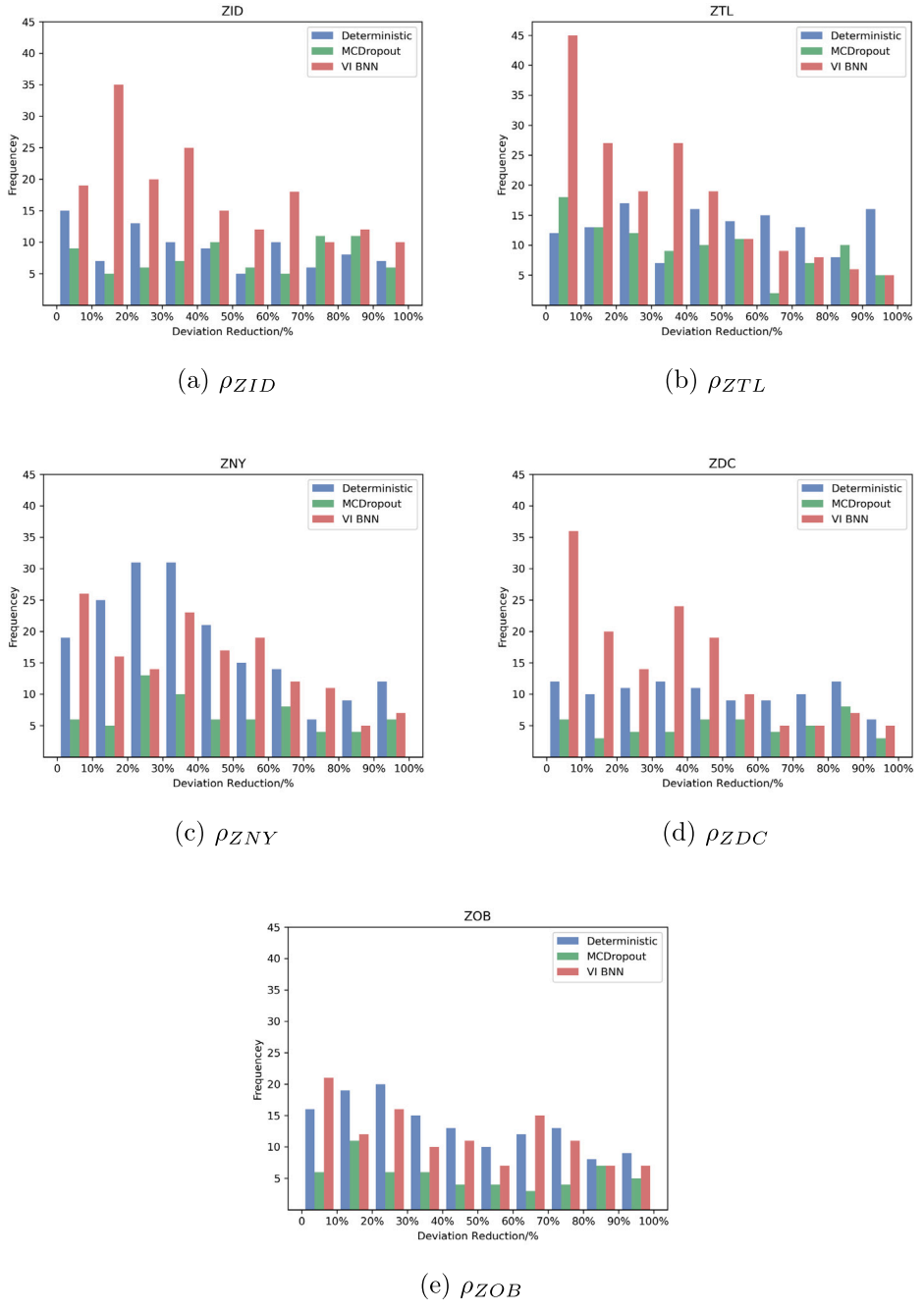
In conclusion, our propose ML pipeline shows considerable TP capabilities. The inclusion of convective weather products exhibits enhancements to the overall prediction power. The variational inference based inference method for Bayesian neural network shows benefit from the aspect of training stability, model regularization, and uncertainty quantification. The deterministic setting also shows effectiveness but with conspicuous deficits.

Furthermore, instead of showing only the deviation reductions based on the mean predictions, we compute the histogram of the variation reduction in ρ. To better understand the deviation reduction capability of the model, we visualize the results for all five flight control sectors in Fig. 8. For example, using VI BNN with the data of ZID, nearly 20 test samples achieve deviation reduction between 0 to 10%. It is obvious that VI BNN model performs better comparing with MCDropout. The VI BNN method is good at generating predictions with lower percentage (e.g. less than 50%) of deviation reductions. And only a few samples have a deviation reduction larger than 50%. The deterministic method also performs well in most of the cases while we are only considering the mean predictions.

6. Conclusion

A major key requirement of NextGen pursuing accurate and reliable aircraft trajectory prediction. To accommodate this, we propose a ground-based strategic trajectory prediction Bayesian neural network with explicit consideration of convective weather products. After the model training, we individually and statistically justify the model performance. A comparison study is performed to evaluate the difference between the deterministic and probabilistic methods. Experiments demonstrate that the VI BNN method shows various benefits for our TP approach, with competitive deviation reduction and complexity. Here is a list of major contributions and conclusions on the proposed study,

- The proposed nested CNN+RNN+FCNN structure is able to handle the complex spatial–temporal correlations of aircraft trajectory and weather conditions. It is scalable and flexible for information fusion and time-series modeling with various data sources and measurements.
- The data processing is critical for successful model learning and data discovery. This includes the proper conduct of data filtering and feature engineering based on domain knowledge expertise, either from the literature or human ATC experts.
- VI is shown to have robust performance compared to the MCDropout for BNN which depends on the subjective dropout ratio selection and requires hyper-parameter search. The deterministic method also shows benefits but lack of uncertainty quantification power.



(a) ρ_{ZID} (b) ρ_{ZTL}

(c) ρ_{ZNY} (d) ρ_{ZDC}

(e) ρ_{ZOB}

Fig. 8. Histogram showing percentage of overall deviation reduction of deterministic, MCDropout, and VI BNN for different sections (a) ZID and (b) ZTL (c) ZNY (d) ZDC (e) ZOB.

- Results show that variance reduction of the proposed method happens for both small and large flight deviation cases. The statistical study on deviation reduction indicates that the proposed method is more likely to reduce less than 50% of the deviations regardless of the choice of control sectors. We notice that the ML model implicitly consider the deviation caused by other factors such as on-board anomalies and pilot's intent cached in the data. Although the prediction deviation can be reduced based purely on the last on-file flight plan, considering the weather explicitly into the ML pipeline can help further reduce the trajectory deviations.

For future work, we have the following suggestions. (a) It would be interesting to investigate the influence of adding other types of weather features, such as vertically integrated liquid (VIL). Also, it would be valuable to consider probabilistic weather forecasting

into the model. All of these can be treated as additional dimensions of the weather feature cubes $X^{weather}$ in this proposed ML pipeline. (b) Other than the weather impact to TP, another impact factor is the traffic management for neighboring aircraft due to potential conflict. This requires the data structure change and search for the nearest neighbor in a dynamic changing spatial–temporal database. In addition, human factors are also critical uncertainty sources as some flights are operated by the visual approach of pilots. (c) The current application focus on the en-route flight and another interesting and important application will be for the near-terminal region, e.g., landing and taking-off phases. Many uncertain events happen near these phases due to increased density, vortex and turbulence effect, and altitude holding patterns. In addition, the standard terminal arrival procedures (STARs) and the standard instrument departure routes (SIDs) based procedures play an important role in near-terminal regions. The consideration of multi-aircrafts awareness is critical and challenging for both the data preparation and algorithm development. For example, the airport surface detection equipment, model X (ASDE-X) data, and the on-aboard Automatic Dependent Surveillance-Broadcast (ADS-B) data can be used for model validation of near-terminal TP with multi-aircrafts awareness. Significant additional research is required for consider aforementioned factors.

CRediT authorship contribution statement

Yutian Pang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Xinyu Zhao:** Conceptualization, Writing – review & editing. **Hao Yan:** Conceptualization, Writing – review & editing. **Yongming Liu:** Conceptualization, Investigation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

The research reported in this paper was supported by funds from NASA University Leadership Initiative program, USA (Contract No. NNX17AJ86 A, PI: Yongming Liu, Technical Officer: Anupa Bajwa). The support is gratefully acknowledged. The authors also would like to thank Dr. Heather Arneson of NASA Ames Aviation Systems Division for the support of this big data research on air traffic management.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S., 2016. Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971.

Ammoun, S., Nashashibi, F., 2009. Real time trajectory prediction for collision risk estimation between vehicles. In: 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing. IEEE, pp. 417–422.

Arneson, H.M., 2018. Sherlock data warehouse.

Avanzini, G., 2004. Frenet-based algorithm for trajectory prediction. J. Guid. Control Dyn. 27 (1), 127–135.

Ayhan, S., Samet, H., 2016. Aircraft trajectory prediction made easy with predictive analytics. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 21–30.

Benavides, J.V., Kaneshige, J., Sharma, S., Panda, R., Steglinski, M., 2014. Implementation of a trajectory prediction function for trajectory based operations. In: AIAA Atmospheric Flight Mechanics Conference. p. 2198.

Besada, J.A., Frontera, G., Crespo, J., Casado, E., Lopez-Leones, J., 2013. Automated aircraft trajectory prediction based on formal intent-related language processing. IEEE Trans. Intell. Transp. Syst. 14 (3), 1067–1082.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D., 2017. Variational inference: A review for statisticians. J. Amer. Statist. Assoc. 112 (518), 859–877.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424.

Bronsvoort, J., McDonald, G., Lopez-Leones, J., Visser, H., 2012. Improved trajectory prediction for air traffic management by simulation of guidance logic and inferred aircraft intent using existing data-link technology. In: AIAA Guidance, Navigation, and Control Conference. p. 4928.

Burden, F., Winkler, D., 2008. Bayesian regularization of neural networks. In: Artificial Neural Networks. Springer, pp. 23–42.

Casado, E., Goodchild, C., Vilaplana, M., 2012. Identification and initial characterization of sources of uncertainty affecting the performance of future trajectory management automation systems. In: Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems, pp. 170–175.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12 (Aug), 2493–2537.

Crisostomi, E., Lecchini-Visintini, A., Maciejowski, J., 2008. Combining monte carlo and worst-case methods for trajectory prediction in air traffic control: a case study. Journal of Guidance, Control and Dynamics 43 (4).

De Leege, A., van Paassen, M., Mulder, M., 2013. A machine learning approach to trajectory prediction. In: AIAA Guidance, Navigation, and Control (GNC) Conference. p. 4782.

DeLaura, R., Robinson, M., Pawlak, M., Evans, J., 2008. Modeling convective weather avoidance in enroute airspace. In: 13th Conference on Aviation, Range, and Aerospace Meteorology. AMS, Citeseer, New Orleans, LA.

Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., Saurous, R.A., 2017. Tensorflow distributions. arXiv preprint arXiv:1711.10604.

Erzberger, H., Lauderdale, T.A., Chu, Y.-C., 2010. Automated conflict resolution, arrival management and weather avoidance for atm.

Erzberger, H., Lauderdale, T., Chu, Y., 2012. Automated conflict resolution, arrival management, and weather avoidance for air traffic management. Proc. Inst. Mech. Eng. G 226 (8), 930–949.

Eshow, M.M., Lui, M., Ranjan, S., 2014. Architecture and capabilities of a data warehouse for atm research. In: 2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC). IEEE, pp. 1E3–1.

FAA, M., 2013. NextGen Implementation Plan. Washington, DC, Available: http://www.faa.gov/nextgen/media/ng2011implementationplan.pdf (Accessed March 18, 2011).

Gal, Y., 2016. Uncertainty in Deep Learning (Ph.D. thesis). University of Cambridge.

Gal, Y., Ghahramani, Z., 2015. Bayesian convolutional neural networks with bernoulli approximate variational inference. arXiv preprint arXiv:1506.02158.
Gal, Y., Ghahramani, Z., 2016a. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. pp. 1050–1059.
Gal, Y., Ghahramani, Z., 2016b. A theoretically grounded application of dropout in recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 1019–1027.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680.
Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A., 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2255–2264.
He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Hong, C., Yu, J., Wan, J., Tao, D., Wang, M., 2015. Multimodal deep autoencoder for human pose recovery. IEEE Trans. Image Process. 24 (12), 5659–5670.
Houenou, A., Bonnifait, P., Cherfaoui, V., Yao, W., 2013. Vehicle trajectory prediction based on motion model and maneuver recognition. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 4363–4369.
Hron, J., Matthews, A.G.d.G., Ghahramani, Z., 2018. Variational bayesian dropout: pitfalls and fixes. arXiv preprint arXiv:1807.01969.
Jones, D.G., Endsley, M.R., 1996. Sources of situation awareness errors in aviation. Aviat. Space Environ. Med..
Kim, B., Kang, C.M., Kim, J., Lee, S.H., Chung, C.C., Choi, J.W., 2017. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 399–404.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Kingma, D.P., Salimans, T., Welling, M., 2015. Variational dropout and the local reparameterization trick. In: Advances in Neural Information Processing Systems. pp. 2575–2583.
Klingle-Wilson, D., Evans, J., 2005. Description of the Corridor Integrated Weather System (CIWS) Weather Products. Project Report ATC-317, MIT Lincoln Laboratory, Lexington, MA.
Knorr, D., Walter, L., 2011. Trajectory uncertainty and the impact on sector complexity and workload. SESAR Innov. Days 29.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105.
Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86.
Lauret, P., Fock, E., Randrianarivony, R.N., Manicom-Ramsamy, J.-F., 2008. Bayesian neural network approach to short time load forecasting. Energy Convers. Manage. 49 (5), 1156–1166.
Le Fablec, Y., Alliot, J.-M., 1999. Using neural networks to predict aircraft trajectories. In: IC-AI. pp. 524–529.
Lee, A., Weygandt, S., Schwartz, B., Murphy, J., 2009. Performance of trajectory models with wind uncertainty. In: AIAA Modeling and Simulation Technologies Conference. p. 5834.
Liu, Y., Hansen, M., 2018. Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. arXiv preprint arXiv:1812.11670.
Liu, W., Hwang, I., 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. J. Guid. Control Dyn. 34 (6), 1779–1789.
Lygeros, J., Prandini, M., 2002. Aircraft and weather models for probabilistic collision avoidance in air traffic control. In: Proceedings of the 41st IEEE Conference on Decision and Control, 2002, Vol. 3. IEEE, pp. 2427–2432.
Lymperopoulos, I., Lygeros, J., 2010. Sequential monte carlo methods for multi-aircraft trajectory prediction in air traffic management. Internat. J. Adapt. Control Signal Process. 24 (10), 830–849.
Lymperopoulos, I., Lygeros, J., Lecchini, A., 2006. Model based aircraft trajectory prediction during takeoff. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. p. 6098.
Manning, C.D., Manning, C.D., Schütze, H., 1999. Foundations of Statistical Natural Language Processing. MIT press.
McNally, D., Sheth, K., Gong, C., Love, J., Lee, C.H., Sahlman, S., Cheng, J., 2012. Dynamic weather routes: a weather avoidance system for near-term trajectory-based operations. In: 28th International Congress of the Aeronautical Sciences. pp. 23–28.
Melis, G., Blundell, C., Kočiský, T., Hermann, K.M., Dyer, C., Blunsom, P., 2018. Pushing the bounds of dropout. arXiv preprint arXiv:1805.09208.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529.
Mondoloni, S., 1998. A genetic algorithm for determining optimal flight trajectories. In: Guidance, Navigation, and Control Conference and Exhibit. p. 4476.
Mondoloni, S., 2006. A multiple-scale model of wind-prediction uncertainty and application to trajectory prediction. In: 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO). p. 7807.
Mueller, T., Sorensen, J., Couluris, G., 2002. Strategic aircraft trajectory prediction uncertainty and statistical sector traffic load modeling. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. p. 4765.
Neal, R.M., 2012. Bayesian Learning for Neural Networks, Vol. 118. Springer Science & Business Media.
Ng, H.K., Grabbe, S., Mukherjee, A., 2009. Design and evaluation of a dynamic programming flight routing algorithm using the convective weather avoidance model. In: AIAA Guidance, Navigation, and Control Conference. p. 5862.
Nilim, A., El Ghaoui, L., Duong, V., Hansen, M., 2001. Trajectory-based air traffic management (tb-atm) under weather uncertainty. In: Proc. of the Fourth International Air Traffic Management R&D Seminar ATM.
Osband, I., Aslanides, J., Cassirer, A., 2018. Randomized prior functions for deep reinforcement learning. In: Advances in Neural Information Processing Systems. pp. 8617–8629.
Paielli, R.A., Erzberger, H., 1997. Conflict probability estimation for free flight. J. Guid. Control Dyn. 20 (3), 588–596.
Pang, Y., Liu, Y., 2020a. Conditional generative adversarial networks (cgan) for aircraft trajectory prediction considering weather effects. In: AIAA Scitech 2020 Forum. p. 1853.
Pang, Y., Liu, Y., 2020b. Probabilistic aircraft trajectory prediction considering weather uncertainties using dropout as bayesian approximate variational inference. In: AIAA Scitech 2020 Forum. p. 1413.
Pang, Y., Yao, H., Hu, J., Liu, Y., 2019. A recurrent neural network approach for aircraft trajectory prediction with weather features from sherlock. In: AIAA Aviation 2019 Forum. p. 3413.
Pearce, T., Zaki, M., Brintrup, A., Anastassacos, N., Neely, A., 2018. Uncertainty in neural networks: bayesian ensembling. Stat 1050, 12.
Phillips, E.H., 1996. Free flight poses multiple challenges. Aviat. Week Space Technol..
Prandini, M., Hu, J., Lygeros, J., Sastry, S., 2000. A probabilistic approach to aircraft conflict detection. IEEE Trans. Intell. Transp. Syst. 1 (4), 199–220.
Qiao, S., Shen, D., Wang, X., Han, N., Zhu, W., 2014. A self-adaptive parameter selection trajectory prediction approach via hidden markov models. IEEE Trans. Intell. Transp. Syst. 16 (1), 284–296.
Ryan, H., Paglione, M., 2008. State vector based near term trajectory prediction. In: AIAA Guidance, Navigation and Control Conference and Exhibit. p. 7144.
Schuster, W., Ochieng, W., 2014. Performance requirements of future trajectory prediction and conflict detection and resolution tools within sesar and nextgen: Framework for the derivation and discussion. J. Air Transp. Manag. 35, 92–101.
Sridhar, B., Sheth, K.S., Grabbe, S., 1998. Airspace complexity and its application in air traffic management. In: 2nd USA/Europe Air Traffic Management R&D Seminar. pp. 1–6.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Stewart, T., Askey, L., Hokit, M., 2012. A concept for tactical reroute generation, evaluation and coordination. In: 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. p. 5586.
Stewart, T., DeArmon, J., Chaloux, D., 2013. Using flight information to improve weather avoidance predictions. In: 2013 Aviation Technology, Integration, and Operations Conference. p. 4377.
Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems. pp. 3104–3112.
Swierstra, S., Green, S., 2003. Common trajectory prediction capability for decision support tools. In: 5th USA/Eurocontrol ATM R&D Seminar. pp. 23–27.
Thipphavong, D., 2008. Analysis of climb trajectory modeling for separation assurance automation. In: AIAA Guidance, Navigation and Control Conference and Exhibit. p. 7407.
Tran, D., Dusenberry, M., van der Wilk, M., Hafner, D., 2019. Bayesian layers: A module for neural network uncertainty. In: Advances in Neural Information Processing Systems. pp. 14633–14645.
Tran, D., Hoffman, M.W., Moore, D., Suter, C., Vasudevan, S., Radul, A., 2018. Simple, distributed, and accelerated probabilistic programming. In: Advances in Neural Information Processing Systems. pp. 7598–7609.
Tran, D., Kucukelbir, A., Dieng, A.B., Rudolph, M., Liang, D., Blei, D.M., 2016. Edward: A library for probabilistic modeling, inference, and criticism. arXiv preprint arXiv:1610.09787.
Wang, Y., Pang, Y., Gorceski, S., Kostiuk, P., Mohen, M.T., Menon, P.K., Liu, Y., 2021. A voice communication-augmented simulation framework for aircraft trajectory simulation. IEEE Trans. Intell. Transp. Syst..
Wen, Y., Vicol, P., Ba, J., Tran, D., Grosse, R., 2018. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. arXiv preprint arXiv:1803.04386.
Wiest, J., Höffken, M., Kreß el, U., Dietmayer, K., 2012. Probabilistic trajectory prediction with Gaussian mixture models. In: 2012 IEEE Intelligent Vehicles Symposium. IEEE, pp. 141–146.
Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c., 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems. pp. 802–810.
Xue, H., Huynh, D.Q., Reynolds, M., 2018. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 1186–1194.
Yepes, J.L., Hwang, I., Rotea, M., 2007. New algorithms for aircraft intent inference and trajectory prediction. J. Guid. Control Dyn. 30 (2), 370–382.
Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S., 2020. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: European Conference on Computer Vision. Springer, pp. 507–523.
Yu, Y., Yao, H., Liu, Y., 2019. Aircraft dynamics simulation using a novel physics-based learning method. Aerosp. Sci. Technol. 87, 254–264.
Zhang, X., Mahadevan, S., 2020. Bayesian neural networks for flight trajectory prediction and safety assessment. Decis. Support Syst. 113246.
Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., Chen, C., 2011. Data-driven intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst. 12 (4), 1624–1639.
Zhu, L., Yu, F.R., Wang, Y., Ning, B., Tang, T., 2018. Big data analytics in intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst. 20 (1), 383–398.
Ziebart, B.D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J.A., Hebert, M., Dey, A.K., Srinivasa, S., 2009. Planning-based prediction for pedestrians. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 3931–3936.