AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1413

Probabilistic Aircraft Trajectory Prediction Considering Weather Uncertainties Using Dropout As Bayesian Approximate Variational Inference

Yutian Pang* , Yongming Liu†
Arizona State University, Tempe, Arizona, 85287, USA

In the context of air traffic management (ATM), an accurate and reliable prediction of the aircraft's trajectory is of critical importance. The enhanced predictability can decrease the chance of flight delays and can detect and reduce safety concerns as earlier stages. Aircraft trajectory prediction (TP) is stochastic in nature and many uncertainty factors will affect the final prediction results, such as weather uncertainties. A novel approach for probabilistic aircraft trajectory prediction is proposed using the Bayesian Neural Network in this paper. This approach has the capability of predicting the aircraft trajectory with the last on-file flight plan prior to takeoff including predictive uncertainties. It's achieved by the use of dropout as Bayesian approximate Variational Inference (VI) in regular neural nets. The experiment is conducted with the Atlanta Air Route Traffic Control Center (ZTL) flight data and the corridor integrated weather system (CIWS) weather data from Sherlock Data Warehouse (SDW) on June 24th, 2019. The model is able to report a confidence interval (CI) of the prediction for both latitude and longitude coordinates. We notice that huge uncertainties still exist in the dataset which requires further investigation of other possible factors.

I. Nomenclature

X, Y	=	training dataset
x^*, y^*	=	testing dataset
ω	=	model parameters
$q_\theta(\omega)$	=	variational parameters
$p(\omega)$	=	weight prior
k	=	row indices of weights
K	=	number of test
ATM	=	air traffic management
TP	=	trajectory prediction
VI	=	variational inference
ZTL	=	atlanta air route traffic control center
SDW	=	sherlock data warehouse
CI	=	confidence interval
BNN	=	bayesian neural networks
CNN	=	convolutional neural networks
RNN	=	recurrent neural network
$LSTM$	=	long short-term memory

II. Introduction

The development of accurate and reliable air traffic trajectory prediction (TP) models is a key objective of the next generation (NextGen) national air transportation system. It is purposed to efficiently and safely accommodate the increasing air traffic flows within the United States Airspace (UAS). It's reported that the demand in air traffic will grow

*Research Associate, School for Engineering of Matter, Transport and Energy, Yutian.Pang@asu.edu.
†Professor, School for Engineering of Matter, Transport and Energy, Yongming.Liu@asu.edu.

by two to three times in the next 25 years [1] which makes the current air transportation system under significant stress. There has been shown to have non-negligible trajectory uncertainties during aviation operations. The uncertainty of trajectory predictions comes from multiple sources. A typical trajectory planning tool makes decision-based on the aircraft's performance, pilot intends and Terminal Radar Approach Control (TRACON) regulations. The unawareness of pilot intend, assumptions to aircraft's conditions, where there is a limit knowledge, results in the uncertainty of the prediction. 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. Furthermore, in addition to these kinds of model uncertainties, environmental factors also cause uncertainty. The convective weather prediction has a certain resolution, accuracy and forecasting interval. However, the convective weather condition usually generates rapidly and poses a significant danger to aviation operations. The adjustments of trajectory coordinate with the weather conditions also result in uncertainties. All of these uncertainties lead to the insufficient, unreliable prediction result of deterministic model output.

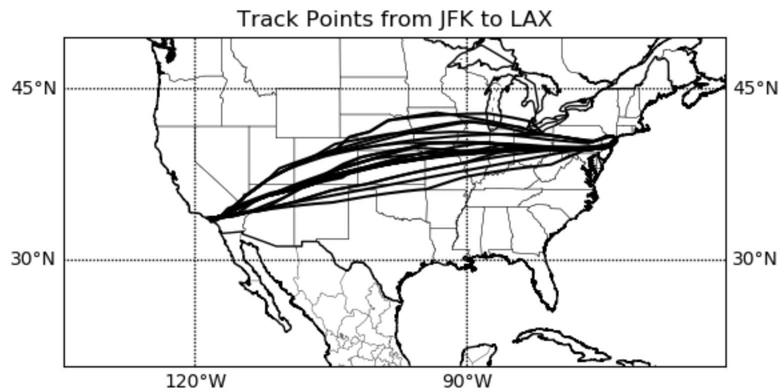


Fig. 1 Flight From JFK to LAX

Fig. 1 shows all the flights flying from John F. Kennedy International Airport (JFK) to Los Angeles International Airport (LAX) within one day. It's clear that the flight tracks deviate from each other. This kind of behavior comes from convective weather conditions, multi-aircraft conflicts and also different airlines that tend to make different flight plans. Trajectory prediction in both spatial and temporal domains, as a popular research topic in the air traffic management (ATM) field, draws the attention of researchers with a different background. Multiple machine learning and deep learning tools in big data analysis fields have been applied to it.

Neural Networks (NN) have gained significant attention during the past decade due to the boom of computation power achieved by GPU clusters. Advances in computer hardware, deep learning models and data availability have enabled the use of deep neural networks into different research fields. It's shown that neural work has achieved great success in object detection [2–4], natural language processing [5], generative learning [6] and reinforcement learning [7]. Meanwhile, the inherent characteristics of NN architecture make are prone to overfitting. This requires tuning of network parameters using a method like Dropout [8]. Moreover, the neural network lacks a proper theoretical theorem as a foundation. Researchers always choose the parameters of the NN model by *best practice* during their research.

Bayesian techniques are suggested as a remedy to address these concerns among society. Bayesian Neural Networks (BNN) is more robust to overfitting problems to control the model complexity [9]. Instead of a single *point estimate* set of model parameters computed from the NN, the Bayesian methods yield distribution for each of the parameters which provide an inherent estimate of prediction uncertainty. The uncertainty, introduced into the BNN by the posterior predictive distribution, is a vital feature to engineering, especially aviation applications. The prior settings in Bayesian approaches also include prior knowledge into NN such that incorporate physical constraints into the process. This will also reduce the amount of data needed for training neural nets by adding prior knowledge. There are also drawbacks to BNNs as well, which is also the reason there are not too much research works in this direction. The main complication is the computational intractability of Bayesian posterior distributions. There have been significant advances in approximate inference, but these are still much more computationally expensive compared to most frequent inference [10].

The objective of the NASA University Leadership Initiative (ULI) project [11] requires researchers to address the safety needs and their technology solutions for future national airspace system (NAS) for both manned and unmanned aircraft [12]. To accommodate this, we developed a Bayesian neural network for non-deterministic aircraft trajectory prediction based on the last on-file flight plan and the convective weather forecasting. The inputs to the model are the flight plan prior to takeoff and weather conditions along with the flight plan. The model output is the predicted flight route with a confidence interval (CI). The scope of this paper is to apply the theory of BNN for aircraft trajectory predicting task, which is a well-defined regression problem. The model is capable of predicting the aircraft trajectory with an uncertainty bound.

This paper is organized as follows: Section II is the brief introduction of Air Traffic Management research as well as Bayesian Neural Network models including the benefits and drawbacks. In section III, we discuss the related work and the development of Bayesian Variational Inference by other researchers. Section IV discusses the data used for evaluation, Bayesian derivation as well as network architecture in this research. The following Section V is a discussion of the experimental results using the data processed and model defined. Lastly, Section VI is the conclusion and comments of this research.

III. Related Work

Researchers have made strenuous efforts on weather-related aircraft trajectory prediction tasks. The related work in trajectory prediction (TP) can be summarized into two categories, the deterministic approach, and the non-deterministic approach.

The deterministic approach stands for the model that predicts the trajectory of a series of points only. As the ATM incorporates machine learning algorithms that gain attention in recent years, there have a few research works doing air traffic prediction based on machine learning algorithms. A model-based aircraft TP during takeoff using the radar measurements of flight tracks which serve as an indicator among candidate trajectories that attend to predict the actual flight data [13]. Similarly, a recent research study applies the Hidden Markov Model (HMM) to predict trajectories taking environmental uncertainties into account [14]. By training the HMM model on a historical trajectory and weather dataset, the author obtained the parameters of HMM. This approach treats the objective airspace as a set of cubes associated with weather parameters as observations of HMM to predict a trajectory among historical trajectory candidates. The above two approaches are deterministic but require the dataset to cover all possible past flight routes for trajectory selection. Different from these selection techniques, there are also research works on prediction techniques that treat the objective as a regression task. Multiple regression algorithms are used by researchers from different backgrounds [15–17]. Another interesting work considers the problem of trajectory deviation caused by the convective weather as the *anomaly* and builds up an analytic classification pipeline for anomaly detection, diagnostics, and prediction [18]. Despite from pure data-driven method, another research work purposed what's called physics-based learning for the trajectory prediction task. The idea is to integrate the underlying physics of aircraft dynamical systems into machine learning models to reduce the training costs and enhance simulation performances [19]. This approach is found to exhibit superior computation efficiency compared with other classical numerical methods.

The non-deterministic approach, or probabilistic approach for aircraft trajectory prediction has been purposed in many research papers. John Lygeros and Maria Prandini purposed a probabilistic model for aircraft position forecasting using the flight plan and current position as input [20]. The model is able to output probability distribution for aircraft positions which will incorporate uncertainty into the predictions. Another work from them [21] purposed a probabilistic model for predicting position in the near-term and mid-term future. They propose 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. Validation of the purposed method is validated using Monte Carlo simulations. Similarly, another trajectory prediction research based on aircraft intent is raised out [22]. This method uses a hybrid estimation algorithm to estimate the aircraft's state and light mode. Then the estimates along with the air traffic region regulations, the flight plan and the environment are used to infer the pilot's intent. Finally, the aircraft trajectory is predicted based on the estimates and the pilot's intent. Another probabilistic trajectory prediction research [23] purposed a stochastic linear hybrid system to describe the dynamics of an aircraft when changing flight modes. The model also incorporates the aircraft's intent information to model the dynamics. The algorithm is computationally efficient and more accurate for aircraft trajectory prediction and conflict detection compared to other methods. The predicted aircraft trajectory is a series of PDFs of the future position. To predict the time of arrival, a generalized linear model (GLM) is developed in the terminal area with the initialization of wind and aircraft state [24]. Despite the traditional methods, machine learning algorithms are incorporated into the ATM field in recent years. A probabilistic trajectory prediction tool with Gaussian Mixture Models (GMM) is developed [25]. Instead of learning of regression function, this method uses previously observed motion patterns to infer a joint probability distribution as the motion model and then predict the trajectory by calculating the probability of the distribution. The model is capable of predicting the position of the object several seconds in advance and also an evaluation of the variance can be used to examine the prediction confidence. Another recent study using a deep generative convolutional recurrent neural network (RNN) approach for 4D trajectory prediction, as of the authors' knowledge, is the first paper using an encoder-decoder recurrent neural structure for this task [26]. 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 filed flight plans and the prediction error metrics show that average absolute horizontal errors are around 50 nautical miles and 2800 feet for average vertical errors.

There are a variety of other research topics in the trajectory prediction area. A sensitivity study on trajectory predictions in ATM introduces the concept of closed-loop sensitivities [27], which are defined as the difference between actual and computed trajectories per unit of modeling errors into pilot feedback controls. This method is applied and validated by the TRACON automation system as well as flight management systems.

IV. Preliminaries

A. Variational Inference

The formulation of model uncertainty relies on probabilistic modeling, Bayesian modeling specifically. If given a training inputs $X = x_1, ..., x_n$ and their corresponding outputs $Y = y_1, ..., y_n$, we try to find the parameters ω for the approximation function $y = f^{\omega}(x)$ that are *most likely* to generate the outputs. The Bayesian approach gives a space of parameters ω as a distribution 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. Bayesian approach also defines a likelihood distribution $p(y|x, \omega)$. The likelihood distribution is a probabilistic model of the model outputs given the data points and model parameters. Then we can calculate predict the output for a new input point x^* by Bayesian inference:

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega)p(\omega|X, Y)d\omega \tag{1}$$

where $p(\omega|X, Y)$ is the *posterior* by Bayesian theorem,

$$p(\omega|X, Y) = \frac{p(Y|X, \omega)p(\omega)}{p(Y|X)} \tag{2}$$

The normaliser in the posterior evaluation $p(Y|X)$ is also called the *model evidence* [28]. In practical, the posterior usually cannot be evaluated analytically. Thus we defined an approximating *variational* distribution $q_\theta(\omega)$ for the posterior. Thus we minimize the Kullback-Leibler (KL) divergence [29] between our approximation and the posterior w.r.t. θ,

$$KL(q_\theta(\omega)||p(\omega|X, Y)) = \int q_\theta(\omega)log\frac{q_\theta(\omega)}{p(\omega|X, Y)}\, d\omega \tag{3}$$

Minimising the KL divergence allows us to approximate the predictive distribution as,

$$q_\theta^*(y^*|x^*) := p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega)q_\theta^*(\omega)\, d\omega \tag{4}$$

This is also equivalent to maximising the evidence lower bound (ELBO) w.r.t. the variational parameters $q_\theta(\omega)$,

$$\mathcal{L}_{VI}(\theta) := \int q_\theta(\omega)\, log(p(Y|X, \omega))\, d\omega - KL(q_\theta(\omega)||p(\omega)) \tag{5}$$

The above procedure is known as Variational Inference (VI) [30] which is a standard technique in Bayesian modeling. The recent advances in VI are used as the fundamental approach to approximate the posterior for deep Bayesian neural nets.

B. Bayesian Neural Network

Bayesian Neural Network (BNN) offers a probabilistic interpretation of deep learning models by inferring distributions over the models' weights as prior knowledge. To train the network given the prior distribution, variational posterior and training data, the loss function is defined as the negative ELBO,

$$\mathcal{L}_{VI}(\theta) := KL(q_\theta(\omega)||p(\omega)) - \mathbb{E}_{q_\theta(\omega)}[log(X,Y|\omega)] \tag{6}$$

The gradient descent can be achieved by minimizing $\mathcal{L}_{VI}(\theta)$. However, the gradient of the two expectations in Eq. (6) cannot be computed explicitly but must be done by Monte Carlo Sampling if the BNN has more than one hidden layer. A backpropagation-like algorithm called *Bayes By Backprop* [31] is obtained for variational Bayesian inference in Neural Networks. *Bayes By Backprop* uses unbiased estimates of gradients of the loss function in Eq. (6) to learn a distribution over the weights of a neural network. The key idea behind it is the fact that under certain conditions, the derivative of an expectation is the expectation of the derivative as Eq. (7). The gradient of the loss function can be evaluated then. The model uncertainty in Bayesian networks is modeled by a technique called *Monte Carlo dropout* [32], which is essentially running several stochastic forward passes through the network and averaging the outputs.

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q_\theta(\omega)}[f(\omega,\theta)] = \mathbb{E}_{q(\epsilon)}[\frac{\partial f(\omega,\theta)}{\partial\omega}\frac{\partial\omega}{\partial\theta} + \frac{\partial f(\omega,\theta)}{\partial\theta}] \tag{7}$$

The recent study on approximate Bayesian VI shows that a neural network with arbitrary depth and non-linearities, with Bernoulli dropout applied before every weight layer, is mathematically equivalent to an approximate variational inference which results in uncertainty estimates [32]. They show that the dropout objective, in effect, minimizes the KL divergence between an approximate distribution and the posterior of a deep Gaussian process (marginalized over its finite rank covariance function parameters).

C. Dropout as Bayesian Approximation

To implement an approximate variational inference to convolutional layers, we place a prior distribution over each kernel and approximately integrate them with Bernoulli variational distributions. Then sample the Bernoulli random variables and multiply with the weight matrix. This approximating distribution is also equivalent to applying the dropout after every convolution layer as well as inner-product layers [28]. Thus the implementation of the Bayesian CNN is therefore as simple as using dropout after every convolution layer before pooling [33]. On the other side, for the recurrent layers, such as Long Short-Term Memory (LSTM) in our work, we follow the implementation of the variational dropout technique called Variational RNN. Implementing the approximate inference is identical to implementing dropout in RNNs with the same network units dropped at each time step, randomly dropping inputs, outputs, and recurrent connections [34]. Compare to the traditional dropout technique applied after RNN cells, variational dropout can drop the connection between each state of the RNN folds. This is implemented by randomly mask each row of the weight matrices ω_k of RNN cell to zero when defining the approximating distribution as,

$$q(\omega_k) = pN(\omega_k; 0, \sigma^2 I) + (1-p)N(\omega_k, m_k, \sigma^2 I) \tag{8}$$

where p is the dropout probability given in advance and m_k is the row vector of variational distributions. σ^2 are small noises added to the weights. The optimization objective is the m_k of the variational parameters of random weight matrices correspond to the standard RNN's weight matrices.

The final Bayesian framework would be a recurrent neural network couple with convolutional layers to extract useful information from the weather features. Variational dropout would be applied to LSTM layers and regular dropout would be applied after every fully-connect and convolutional layers. The prediction of the model and prediction uncertainties can be approximated by approximating the posterior in Eq.(9),

$$p(y^*|x^*, X, Y) \approx \int p(y^*|x^*, \omega)q(\omega)d\omega \approx \frac{1}{K}\sum_{k=1}^{K} p(y^*|x^*, \hat{\omega_k}) \tag{9}$$

which is just performing dropout at each test time and averaging results of K tests. x^* and y^* stand for the test dataset. X and Y is the training dataset. ω is the model weights we optimized. This procedure is also called Monte Carlo dropout (*MCdropout*).

V. Methodology

A. Data Preparation

We use control sector flight data for our experiment instead of the whole flight tracks from departure airport to the arrival airport. Each of the control sector (ARTCC) is in charge of all of the flights flying within the airspace and each control sector may have a certain control behavior. Thus the prediction across multiple control sectors is not the best approach. The sector-based TP means our trained model is sector-specific and can handle flight tracks within the airspace of the current control sector. We choose to download the data on June 24th, 2019 since it is reported to have severe weather conditions that happened on this day*. The typical severe weather conditions we are looking at are tornados, large hail, and strong wind conditions. The record shows that strong wind and large hail is reported multiple times within the range of Atlanta Air Route Traffic Control Center (ZTL). Thus we are focusing on predicting the flight trajectories in ZTL on June 24th, 2019.

The data used for this research are obtained from the Sherlock Data Warehouse (SDW) [35]. Sherlock is a distributed big data platform for data visualization to support air traffic management (ATM) research, which includes a database, a web-based user interface (UI), a few data visualization tools and other services. It is a platform for reliable ATM data collection, archiving, processing, query, and delivery and can be used for big data analysis, including data mining and machine learning [35]. Data of Sherlock comes primarily from the federal aviation administration (FAA) and the National Oceanic Atmospheric Administration (NOAA) [36]. There are multiple sources feeding into Sherlock, such as flight plan and track from ARTCC, Rapid Refresh (RR) Weather Forecast from NOAA including wind, temperature and pressure, current and forecast precipitation and echo tops from Convective Integrated Weather Service (CIWS) and FAA System Wide Information Management (SWIM) data sources for flight data. SDW will parse, process the raw data from the source and store them into a data repository. The approved user is able to query and download the integrated data by specifying a date. Here we only use the data from two sources. The Integrated Flight Format (IFF) flight data from SWIM and Echo Top (ET) convective weather data from CIWS. The processing procedure of weather data stays the same as our previous work. However, the parser for flight data has some minor differences.

1. IFF Flight Data

The flight data is stored as Integrated Flight Format (IFF). It includes all of the source raw 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. It can be translated into WGS84 coordinates using the web-based database. We download the IFF data for ZTL on June 24th, 2019 and process the flight plan and track points to make them equal length. We first take out the flight tracks from the sector IFF file, then do linear interpolation to each of the track points with a 1-second interval. Then match the parsed string format of flight plan with the interpolated flight tracks and remove the flight waypoint that stays outside the sector track ranges. We use Euclidean distance to evaluate whether waypoints belong to the current sector range. Then do the same interpolation to the flight plan to make it has the equal length as the track points. Lastly, we equally sample data points from these two sequences as the processed flight tracks and processed track points used for our model. The length of the processed data is set to be 50 but can be changed to any other value. We've found 6255 sequences of flight records from one day's IFF file and plot it here as shown in Fig. 2. The plots also verify the location of the tracks lies in the range of ZTL. Only longitude and latitude coordinates are considered in this experiment thus the dimension of the final processed flight tracks and flight plans are $6255 \times 50 \times 2$.

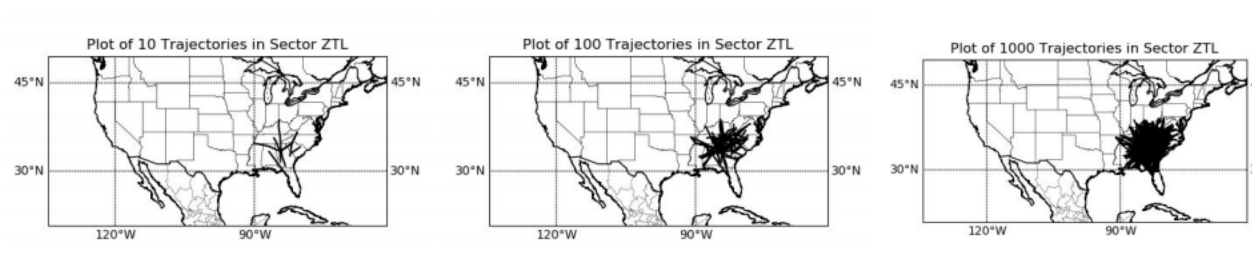


Fig. 2 Flights in ZTL on June 24th, 2019

* Wind and Tornado reported based on National Weather Service Storms Prediction Center `https://www.spc.noaa.gov/exper/archive/event.php?date=20190624`

2. Weather Data

The weather data are obtained from Corridor Integrated Weather Systems (CIWS) in SDW. CIWS acquires data from terminal weather sensing systems, and National Weather Service sensors and forecast products, and automatically generate convective weather products for display on existing systems in both terminal and en route airspace within the CIWS domain. The two key features of CIWS, Echo Top (ET) and Vertically Integrated Liquid (VIL), both include the current and forecast dataset in SDW. In this research, we only use the current ET data but it can be applied to the forecast dataset without too much effort. At each given track points, we take out a weather cube ahead of the current location from the original weather file and rotate it with the heading angle of the aircraft. This is the same weather cube generator used in our previous work. Different from the previous setting, we change the cube size to be 32×32 after scaling the original resolution of weather files by 10 times. The width of the weather band is able to cover sufficient area under this setting. Thus the final dimension of the weather feature cubes is $6255 \times 49 \times 32 \times 1$.

Thus the final processed data includes three parts, the flight plan tensor p as the initial tracks to be calibrated, the weather tensor w as the cause of uncertainties to the trajectory and the true trajectory as the ground truth of the prediction to be a catch with. Each of the pairs of these three is considered as one data point. The total data point we have in our work is 6255. The dimension of flight plan and true trajectory is 50×2 and the dimension for the weather tensor is $49 \times 32 \times 1$.

B. Network Architecture

The network model includes two three dimensional convolutional layers, one RNN cell and a few fully-connect (dense) layers. The convolutional layers are used to extract useful features from w into a vector. The extracted weather vector will be concatenated with the flight plan tensor p as the inputs to the RNN cell. The prediction is just the output from multiple dense layers. This framework is also shown in Table. 1. In this table, N represents the batch size and empty cell means not applicable. The second figure in the tensor dimensions (49 in this case) is the length of the sequence (tracks). We use the first point of the sequence as the initial thus only the last 49 points are used in the model.

Table 1 Network Architecture

Layers	Input Size	Output Size	Dimension	Others
Conv3d_1	[N, 49, 32, 32, 1]	[N, 49, 10, 10, 3]	[1, 5, 5, 3]	Strides: [1, 3, 3, 1], No Padding
Conv3d_2	[N, 49, 10, 10, 3]	[N, 49, 4, 4, 1]	[1, 3, 3, 1]	Strides: [1, 3, 3, 1], Zero Padding
Flatten	[N, 49, 4, 4, 1]	[N, 49, 16]		
Dense_1	[N, 49, 16]	[N, 49, 4]	4	
Dense_2	[N, 49, 4]	[N, 49, 1]	1	
Concat	[N, 49, 1]	[N, 49, 3]		Concatenate with Flight Plan p
LSTM	[N, 49, 3]	[N, 49, 128]	128	Variational Dropout
Dense_3	[N, 49, 128]	[N, 49, 64]	64	
Dense_4	[N, 49, 64]	[N, 49, 32]	32	
Dense_5	[N, 49, 32]	[N, 49, 2]	2	No Dropout

The initial input to the model is the weather cube tensor w 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 $[1, 3, 3, 1]$ and a stride of $[1, 3, 3, 1]$ and zero padding, respectively. The output has a size of $[N, 49, 16]$ after flatten 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 extracted. 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 128. 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 model loss function is defined as the standard Root Mean Squared Error (RMSE) of the prediction and the ground truth.

The use of dropout operation can be summarized as applying regular dropout layers after each convolutional and dense layers, and perform variational dropout to LSTM cell as described in Yarin Gal's paper [32]. It worth pointing out that there is no dropout operation after layer Dense_5 since the hidden dimension is 2.

The training process is performed on a workstation with Intel Xeon E5-1620 v4 @3.50 GHz CPU and a single NVIDIA GTX 1080 GPU running tensorflow-gpu 1.6.0[†]. The data is separated into a training set and a testing test with a weight of 0.8 and 0.2. Normalization is performed before training. We choose to use momentum optimizer with a parameter of 0.9. We set the weight decay of learning rate following Eq. (10),

$$decay\_lr = \frac{learning\_rate}{(1 + \frac{decay\_rate * global\_step}{decay\_step})^r} \tag{10}$$

with the initial of learning rate is $1e^{-3}$ and the decay rate is $1e^{-4}$ with the power r equals to 0.75 and global steps of 1. The dropout ratio of all layers during both training and testing is set to be 0.5.

VI. Experimental Results

As we described in the previous section, the final prediction result is the output of the sequential model when minimizing the RMSE between the prediction and the ground truth. The inputs to the mode are flight plan tensor p, weather tensor w and the ground truth. We've obtained the experimental results after running 4000 epochs of the training data. The prediction means and confidence bound is determined after running 100 MCdropout tests. During each test, the dropout operation will randomly mask the weight of each parameter to zero with a probability of the dropout ratio we defined. Fig. 3 is the plot of two out-of-sample testing results. The Left three plots belong to the 65th test data for different confidence intervals and the right plots are for the 92nd data, respectively. To better visualize the confidence range of each prediction point, we draw out the confidence ellipses, in consideration of both the longitude and latitude coordinates are random variables. The axis of the ellipses is determined by the standard deviation of the many tests. The ellipses are rotated along the direction of the flight trajectories. The plot range is set to be the range of sector ZTL. Ideally, we would like to make the red dots to be as close as possible to the green dots. We notice that the prediction interval is capable of capture the true trajectory although the predicted tracks are not optimal. The prediction interval is acceptable compared to our previous work with generative nets. However, the model is not able to capture the last few data points near the terminal but output relevant large uncertainty. The terminal area control is more complicated which is outside the scope of this research.

$$L2_k^{ori} = \sum_i^n \sum_j^d (Y_{k,i,j}^{true} - Y_{k,i,j}^{p})^2 \tag{11}$$

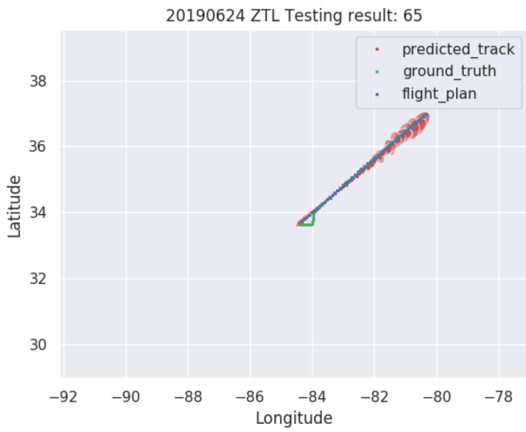
$$L2_k^{new} = \sum_i^n \sum_j^d (Y_{k,i,j}^{true} - Y_{k,i,j}^{pred})^2 \tag{12}$$

$$reduction = \frac{Var(L2_k^{ori}) - Var(L2_k^{new})}{Var(L2_k^{ori})} \tag{13}$$

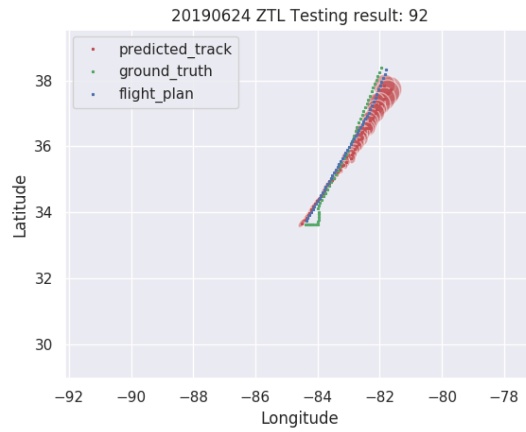
We also report the statistical analysis results here based on the criterion we defined. The value we used is the mean prediction value from many tests. We define the L_2 norm as the evaluation metric of prediction errors. The quantities we are interested in are the difference between the original flight plan and true tracks and the difference between the model prediction and the true tracks. The calculation is shown in Eq. (11) and Eq. (12). The parameter n here stands for the fold number (50 in this case) and d stands for the dimensions of the inputs. Typically, d for longitude and latitude prediction only is 2. Eq. (13) is the equation we used to calculate the overall variance reduction.

We've listed the results from three different deterministic models in Table. 2. The prediction power of the Bayesian approach is worse than the other two methods. The reason is the averaged loss for the Bayesian approach is not able to reach a lower value than the other two methods during training because of the use of dropout. However, the generation power to output uncertainties is significantly improved compared to the generative nets. The randomness of the Bayesian approach comes from the dropout of weights instead of the randomly sampled noise input to the network of generative nets. In other words, the model obtains generative power while losing the prediction accuracy for this Bayesian approximate variational inference approach. The causal inference of variational inference is believed to

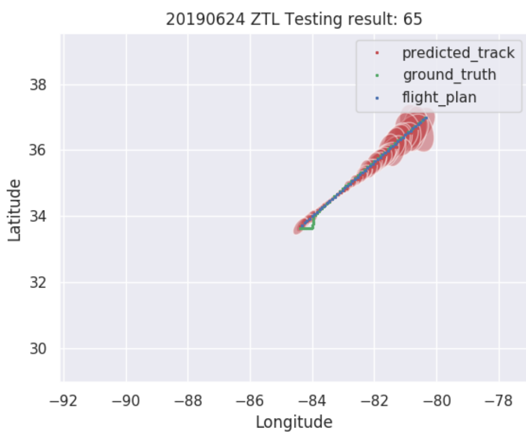
[†]The data parsers and weight decay functions can be found at `https://github.com/YutianPang/Weather-Avoidance/tree/master/sherlock_sector_parser`



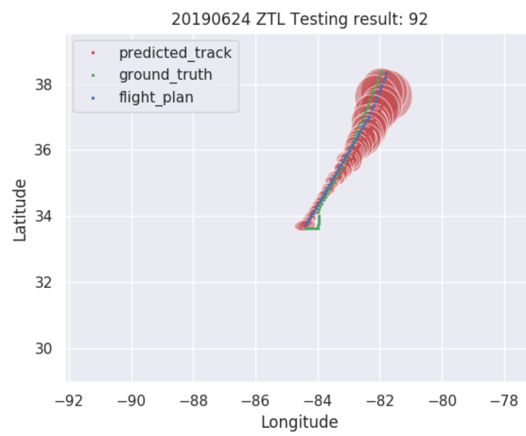
(a) 2σ interval



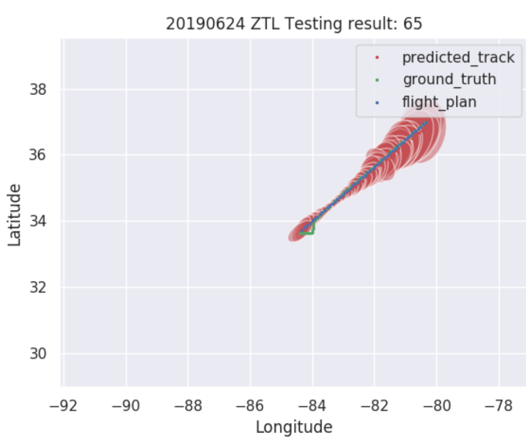
(b) 2σ interval



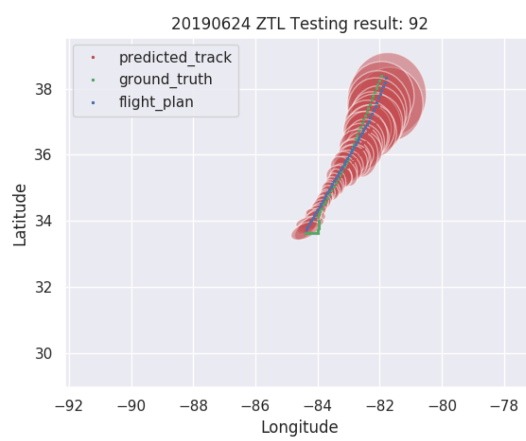
(c) 4σ interval



(d) 4σ interval



(e) 6σ interval



(f) 6σ interval

Fig. 3 Plot of two testing results with three different confidence interval

overcome this issue which is one of the future work on non-deterministic neural nets. We also want to point out the issue with the dataset. We notice that a large amount of the data with flight deviations doesn't have the weather-related pattern. This leads to the difficulty in our work significantly. To address this, we will first use simulated data for model validation in the future.

Table 2 Comparison to Previous Work

Models	Percentage of Flights Reduced	Overall Variance Reduction
Conv-LSTM	47.0%	12.3%
Generative Model	55.2%	22.1%
Dropout as Bayesian	26.2%	16.8%

VII. Conclusion

To address the safety concerns required by the NextGen, we developed a Bayesian neural network for non-deterministic aircraft trajectory prediction based on the last on-file flight plan and the convective weather forecasting. The Bayesian framework follows the definition of using dropout as Bayesian approximate variational inference. The model is composed of two three dimensional convolutional layers, one RNN cell, and a few dense layers. Regular dropout is performed to convolutional layers and dense layers and variational dropout is used for the RNN cell. The optimization objective is defined as the RMSE between the prediction and the ground truth (true trajectory). The experiment is conducted with the sector ZTL flight data and weather data from Sherlock Data Warehouse on June 24th, 2019. The model is able to predict the aircraft trajectory as well as outputting an uncertainty bound when performing MCdropout during testing. The uncertainty bound can capture the ground truth given a certain confidence value compare to our previous work. This dropout as Bayesian framework can be used to generate uncertainties for any given network model easily.

Acknowledgments

The work related to this research was performed at the Prognostic Analysis and Reliability Assessment Lab at Arizona State University. The research reported in this paper was supported by funds from NASA University Leadership Initiative program (Contract No. NNX17AJ86A, Project Officer: Dr. Anupa Bajwa, Principal Investigator: Dr. Yongming Liu). The support is gratefully acknowledged. We also want to thank Dr. Hao Yan from Arizona State University for helpful suggestions on this work.

References

[1] Swenson, H., Barhydt, R., and Landis, M., "Next generation air transportation system (ngats) air traffic management (atm)-airspace project," Tech. rep., Technical report, National Aeronautics and Space Administration, 2006.

[2] Girshick, R., "Fast r-cnn," *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[3] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, 2015, pp. 91–99.

[4] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask r-cnn," *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[5] Manning, C. D., Manning, C. D., and Schütze, H., *Foundations of statistical natural language processing*, MIT press, 1999.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] 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., "Human-level control through deep reinforcement learning," *Nature*, Vol. 518, No. 7540, 2015, p. 529.

[8] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, Vol. 15, No. 1, 2014, pp. 1929–1958.

[9] Lauret, P., Fock, E., Randrianarivony, R. N., and Manicom-Ramsamy, J.-F., "Bayesian neural network approach to short time load forecasting," *Energy conversion and management*, Vol. 49, No. 5, 2008, pp. 1156–1166.

[10] Back, A., and Keith, W., "Bayesian Neural Networks for Financial Asset Forecasting," , 2019.

[11] Liu, Y., and Goebel, K., "Information Fusion for National Airspace System Prognostics," *PHM Society Conference*, Vol. 10, 2018.

[12] Hu, J., "Probabilistic Risk Assessment and Mitigation for UAS Safety and Traffic Management," *Proceedings of the Annual Conference of the PHM Society*, Vol. 11, 2019.

[13] Lymperopoulos, I., Lygeros, J., and Lecchini, A., "Model based aircraft trajectory prediction during takeoff," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6098.

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

[15] Pang, Y., Yao, H., Hu, J., and Liu, Y., "A Recurrent Neural Network Approach for Aircraft Trajectory Prediction with Weather Features From Sherlock," *AIAA Aviation 2019 Forum*, 2019, p. 3413.

[16] Pang, Y., Xu, N., and Liu, Y., "Aircraft Trajectory Prediction using LSTM Neural Network with Embedded Convolutional Layer," *Proceedings of the Annual Conference of the PHM Society*, Vol. 11, 2019.

[17] Wang, Y., Pang, Y., Liu, Y., Dutta, P., and Yang, B.-J., "Aircraft Trajectory Prediction and Risk Assessment Using Bayesian Updating," *AIAA Aviation 2019 Forum*, 2019, p. 2936.

[18] Zhao, X., Yan, H., Li, J., Pang, Y., and Liu, Y., "Spatio-temporal Anomaly Detection, Diagnostics, and Prediction of the Air-traffic Trajectory Deviation using the Convective Weather," *Proceedings of the Annual Conference of the PHM Society*, Vol. 11, 2019.

[19] Yu, Y., Yao, H., and Liu, Y., "Aircraft dynamics simulation using a novel physics-based learning method," *Aerospace Science and Technology*, Vol. 87, 2019, pp. 254–264.

[20] Lygeros, J., and Prandini, M., "Aircraft and weather models for probabilistic collision avoidance in air traffic control," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, Vol. 3, IEEE, 2002, pp. 2427–2432.

[21] Prandini, M., Hu, J., Lygeros, J., and Sastry, S., "A probabilistic approach to aircraft conflict detection," *IEEE Transactions on intelligent transportation systems*, Vol. 1, No. 4, 2000, pp. 199–220.

[22] Yepes, J. L., Hwang, I., and Rotea, M., "New algorithms for aircraft intent inference and trajectory prediction," *Journal of guidance, control, and dynamics*, Vol. 30, No. 2, 2007, pp. 370–382.

[23] Liu, W., and Hwang, I., "Probabilistic trajectory prediction and conflict detection for air traffic control," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, 2011, pp. 1779–1789.

[24] De Leege, A., van Paassen, M., and Mulder, M., "A machine learning approach to trajectory prediction," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 4782.

[25] Wiest, J., Höffken, M., Kreßel, U., and Dietmayer, K., "Probabilistic trajectory prediction with Gaussian mixture models," *2012 IEEE Intelligent Vehicles Symposium*, IEEE, 2012, pp. 141–146.

[26] Liu, Y., and Hansen, M., "Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach," *arXiv preprint arXiv:1812.11670*, 2018.

[27] Jackson, M. R., Zhao, Y. J., and Slattery, R. A., "Sensitivity of trajectory prediction in air traffic management," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, 1999, pp. 219–228.

[28] Gal, Y., "Uncertainty in deep learning," Ph.D. thesis, PhD thesis, University of Cambridge, 2016.

[29] Kullback, S., and Leibler, R. A., "On information and sufficiency," *The annals of mathematical statistics*, Vol. 22, No. 1, 1951, pp. 79–86.

[30] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K., "An introduction to variational methods for graphical models," *Machine learning*, Vol. 37, No. 2, 1999, pp. 183–233.

[31] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D., "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[32] Gal, Y., and Ghahramani, Z., "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *international conference on machine learning*, 2016, pp. 1050–1059.

[33] Gal, Y., and Ghahramani, Z., "Bayesian convolutional neural networks with Bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[34] Gal, Y., and Ghahramani, Z., "A theoretically grounded application of dropout in recurrent neural networks," *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[35] Arneson, H. M., "Sherlock Data Warehouse," 2018.

[36] Eshow, M. M., Lui, M., and Ranjan, S., "Architecture and capabilities of a data warehouse for ATM research," *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, IEEE, 2014, pp. 1E3–1.