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

10.2514/6.2020-1853

Conditional Generative Adversarial Networks (CGAN) for Aircraft Trajectory Prediction considering weather effects

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

The development of air traffic trajectory prediction models is a key objective of the next generation (NextGen) national air transportation system. Significant uncertainties associated with weather conditions exist in the current trajectory prediction methods and should be properly addressed. This paper proposes a novel Conditional Generative Adversarial Network (CGAN) approach for weather-related aircraft trajectory prediction problems. The problem is formulated as predicting the trajectory and conditioning on the last on-file flight plan and weather effects. The generator network includes two convolutional layers and two dense layers for weather feature extraction. Then the extracted features, concatenating with the conditional inputs are feed into a single layer long short-term network to output the generated trajectory. The discriminator network has a similar architecture but is trying to discriminate the inputs from the ground truth dataset and the generated trajectory. The experiment is conducted with the data obtained from Sherlock Data Warehouse (SDW). We are using the flight data from Indianapolis Air Route Traffic Control Center (ZID) and the EchoTop (ET) weather features within the sector airspace on June 24th, 2019. The experimental result shows the mean prediction has a better overall variance reduction compared to our previous work. The model is able to output confidence intervals (CI) of the prediction by sampling random noise as the inputs to the generator.

I. Nomenclature

FAA	=	federal aviation administration
$CIWS$	=	corridor integrated weather system
ET	=	echo top
ATM	=	air traffic management
NWS	=	national weather service
DWR	=	dynamic weather rerouting
TP	=	trajectory prediction
VAE	=	variational autoencoders
$CGAN$	=	conditional generative adversarial network
CNN	=	convolutional neural network
SDW	=	sherlock data warehouse
$NOAA$	=	national oceanic atmospheric administration
IFF	=	integrated file format
RNN	=	recurrent neural network
$LSTM$	=	long short-term memory
FP	=	flight plan
VIL	=	vertically integrated liquid
NAS	=	national airspace system
$Seq2Seq$	=	sequence to sequence
HMM	=	hidden markov model
$RMSE$	=	root mean squared error
CI	=	confidence interval
$ARTCC$	=	air route traffic control center

*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, Senior Member AIAA.

II. Introduction

The next-generation national air transportation system (NextGen[1]) includes several improvements such as integration of the data coming from the existing myriad of aviation data sources, system-level trajectory planning, dynamic weather reroutes (DWR) and multi-aircraft conflict resolution. It is a key research subject among universities and research centers around the country [2] during the past few decades. As a major part of the NextGen's research, the aircraft trajectory prediction (TP) task under weather-related uncertainties contributes a lot to it. Strategic Trajectory Prediction (TP) in both temporal and spatial space is a popular research topic in aviation society. The use of ground-based TP tools has been applied to air traffic controllers with traffic management [3], efficient runway utilization [4], conflict detection [5] and also on weather avoidance predictions [6, 7].

The work on TP can be separated into the deterministic approach and the non-deterministic approach based on whether the model performs a *point estimate* of coordinates or not. Most of the work on TP belongs to the deterministic prediction methods. 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 [8]. Similarly, another recent research study applies the Hidden Markov Model (HMM) to predict trajectories taking environmental uncertainties into account [6]. 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. In recent years, deep learning has achieved great success and drawn huge attention to the community. The use of machine learning/deep learning techniques into the air traffic management (ATM) field increased significantly. A research 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 [9]. 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 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 2,800 feet for average vertical errors.

Generative Adversarial Network (GAN) [10], as a competitor of variational autoencoders (VAE), has been shown that could produce realistic fake images of faces [11], chairs [12] and animals [13]. The generator and the discriminator form the architecture of GAN. The generator produces fake data and outputs a value that indicates the chance that the output is a ground truth data. The goal is to maximize the chance to recognize the generated data as real data by the discriminator. While the loss from the discriminator represents the difference between the model prediction and the ground truth. Thus we want to minimize the discriminator loss as much as possible. This makes the GAN a mini-max optimization problem. GAN is a good method for semi-supervised learning. It shows effectiveness when there are few labeled data samples. The limitations and difficulties in getting the data have shown to have negative effects on the model predictions [14, 15]. The architecture of GAN, with a generator to generate data from random samples, could intrinsically alleviate the issue with limited data. The formulation of condition generative adversarial network (CGAN) [16] is as simple as adding additional inputs to both the generator and the discriminator. Research works on CGAN has shown it can perform regression tasks from simple datasets [17] to complex regression tasks [18] like melody generator from lyrics. Another major benefit of using GAN is it introduces uncertainties into the prediction thus makes it a non-deterministic approach of prediction. This is achieved by sampling different random noises to the generator and perform the multiple tests to the testing dataset. Also, the generated artificial data is highly aligned with the ground truth data compared with VAE since the objective of GAN makes it indistinguishable by the discriminator, which is also a neural network. On the contrary, research has complained about the difficulty and instability during the training process of GAN. The gradient descent algorithm has shown drawbacks to find the equilibrium of the objective functions. A detailed discussion of GAN will be stated in the later session.

As part of the NASA University Leadership Initiative (ULI) project [19] which aims to address the safety needs and their technology solutions for future national airspace system (NAS), we develop a novel GAN architecture to predict the aircraft trajectory prior departing using the last on-file flight plan and convective weather information in order to reduce the weather-related safety uncertainties and output a calibrated flight route. Our main contribution in this paper is to develop a simple and effective GAN architecture and training strategy that could predict the aircraft trajectory as well as given a confidence interval (CI) of the prediction under certain weather uncertainties.

This paper is organized as follows: Section II is the introduction of TP models and the related work done by other researchers in the past few decades. Section III briefly introduces the formulation of CGAN, its achievements and model setup. Section IV discusses the data preparation process and model architecture used in this work in detail. The

following Section V is a discussion of the experimental results. Section VI is the conclusion of this research work.

III. Preliminaries

A. Conditional Generative Adversarial Nets

GAN is proposed by *Ian J. Goodfellow* in 2014 [10]. It's an adversarial framework used to estimate the generative models. It is composed of two deep neural nets, the generator G, and the discriminator D. As a kind of deep generative model, which mainly used for approximating intractable probabilistic computations, GAN is also pitted against the adversarial discriminator. Adversarial approaches have the advantage that there is not a statistical inference process and Markov chains. The generator tends to produce generated data to imitate the ground truth and use it without detection while the discriminator is trying to determine whether the data is from the generated set or the ground truth set. The competition between the generator and the discriminator forces them to keep improving their methods to make the counterfeits indistinguishable from the ground truth data. We are training the generator to create data that towards the discriminator believes it is the ground truth. If we define a prior noise variable z, the ground truth set x then D and G are playing a mini-max game in the sense of cross-entropy loss $V(D, G)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)}[log(D(x))] + \mathbb{E}_{x \sim P_z(z)}[log(1 - D(G(z)))] \tag{1}$$

The $D(x)$ represents the probability that the discriminator believes x is from the ground truth instead of the generator. The back-propagation process of D and G are performed separately. It is also worth pointing out that the ratio of updating steps between G and D plays a significant role in finding the equilibrium. *Goodfellow* also proves Eq. (1) has a global optimum when $P_g = P_{data}$ which requires P_g converges to P_{data}. The concept of generative nets framework is first raised in [10] but the idea of having two neural networks compete comes from the predictability minimization work [20] where the weights from one neural net are trained to predict the weights of another neural net. A famous research work called SeqGAN [21] is purposed recently to learning sequential data using GAN with a policy gradient.

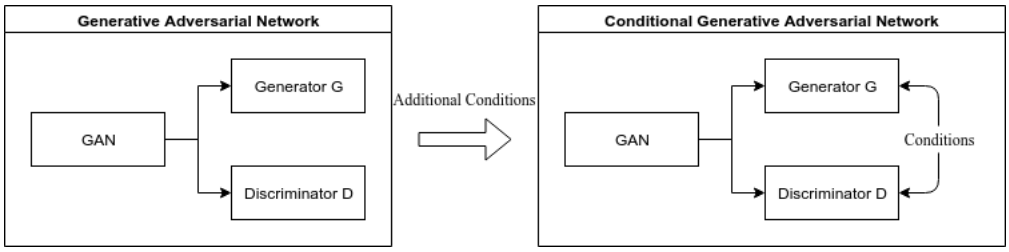

Fig. 1 The transformation from GAN to CGAN

As the original GAN paper also points out in the future work section, a conditional generative model can be obtained by adding additional inputs to both G and D as Fig. 1 shows out. The conditional model can control the data being generated [16]. The additional information is able to direct the data generation process. The additional information y can be any kind of extra information and the objective would be modified as Eq. (2).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)}[log(D(x|y))] + \mathbb{E}_{x \sim P_z(z)}[log(1 - D(G(z|y)))] \tag{2}$$

There is tremendous progress towards the applications of GAN and CGAN architectures in image processing applications. However, there is no application to regression tasks, which is generating outputs from a small dimension output space, until a recent study on regression with CGAN using synthetic datasets [17]. They generate the output space implicitly using $y = f(x, z)$ where z is a specified prior distribution and f is the deep nets model. The result shows that CGAN possesses a better ability to model complex noise forms and is competitive with state-of-art methods. Another research work on melody generation with lyrics [18] with a Conditional LSTM-GAN architecture shows that the model can infer plausible and tuneful melody sequences from the lyrics. They are using a similar deep LSTM generator-discriminator architecture as our work but without convolutional layers.

In our proposed model, the conditions are the last on-file flight plan p and the weather conditions along with the flight plan w. The ground truth x would be the true trajectory. The prior random noise z is defined as a truncated normal

distribution. The model setup can be concluded as,

$$y = f(x, z|p, w) \tag{3}$$

The preparation of each part of the data will be discussed in the later section in detail. We use the Root Mean Squared Error (RMSE) as the evaluation metrics of the model.

IV. Methodology

A. Data Preparation

We decide to use the sector flight data in this work instead of the whole trajectory from one airport to another destination 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 happened on this day. Fig. 2 is a plot from NOAA's Storm Prediction Center*. The red dots represent there is tornado reported and the blue dots show the location where the high wind is reported.

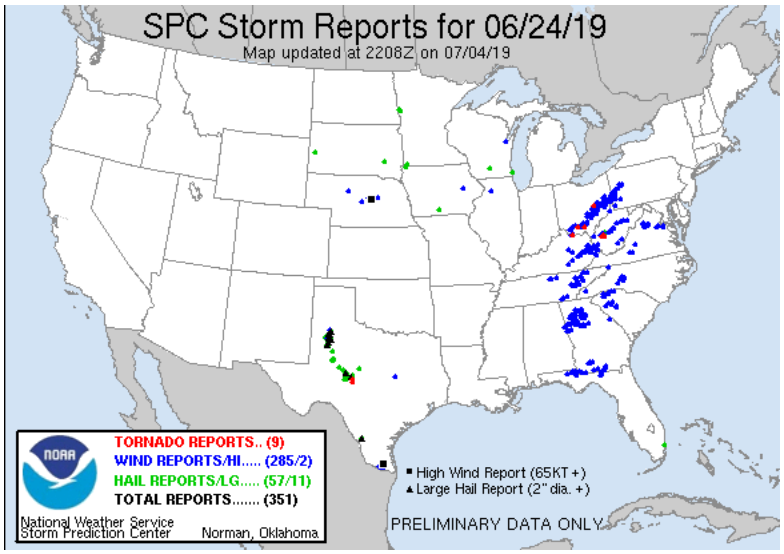

Fig. 2 SPC Storm Reports for 06/24/19

The data used for this research are obtained from the Sherlock Data Warehouse (SDW) [22]. 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 [22]. Data of Sherlock comes primarily from the federal aviation administration (FAA) and the National Oceanic Atmospheric Administration (NOAA) [23]. 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 figure is taken from https://www.spc.noaa.gov/exper/archive/event.php?date=20190624

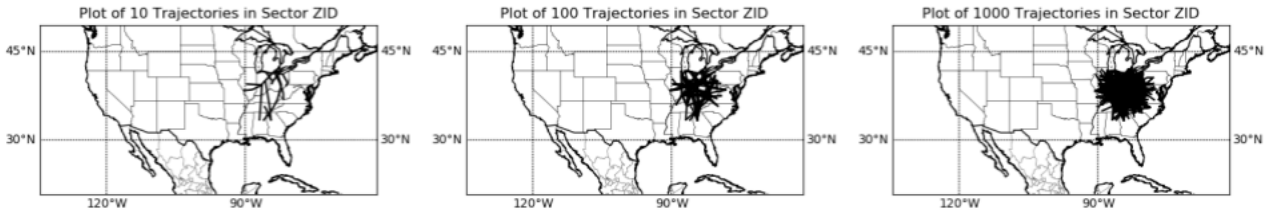

Fig. 3 Flights in ZID on 06/24/2019

1. Sector Flight Data

The flight data is collected from different FAA facilities and stored as IFF. It includes all 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. We developed a web-based [24] tool to translated it into WGS84 coordinates[†]. We choose Indianapolis Air Route Traffic Control Center (ZID) for our demonstration since most of the tornado and high winds are reported around this area. Fig. 3 visualizes the flight tracks flying inside sector ZID and the position corresponding with the position that has severe weather conditions.

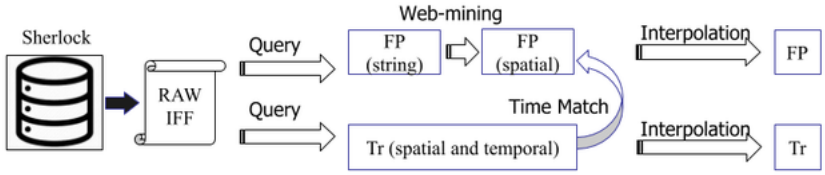

Fig. 4 Data Processing Flow Chart

Fig. 4 shows the procedure of processing sector flight data. 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. We've got 4103 flights after processing the flight data.

2. Weather Data

The weather data are obtained from Corridor Integrated Weather Systems (CIWS) in SDW. CIWS is designed to improve convective weather decision support for congested en-route airspace. It acquires data from FAA 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 [25]. The two key features of CIWS, Echo Top (ET) and Vertically Integrated Liquid (VIL), both come with current and forecast dataset in Sherlock.

In this research, we only use the current ET data as it's more holistic than the forecast 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 circumstance. Only longitude and latitude coordinates are considered in this experiment. Thus the final dimension of the weather feature cubes is $4103 \times 49 \times 32 \times 1$ and the dimension of the flight tracks and flight plans are $4103 \times 49 \times 2$.

[†]All of the data parsers used can be found at *https://github.com/YutianPang/Weather-Avoidance/tree/master/sherlock_sector_parser*

3. Random Noise

The architecture of CGAN also requires a random variable as the input of the generator. We use truncated normal distribution $z \sim N(0.5, 1.0)$ with a lower bond of 0 and the upper bond of 1. During training, the data is randomly sampled from the distribution with the same size as the flight tracks and flight plans.

B. Network Architecture

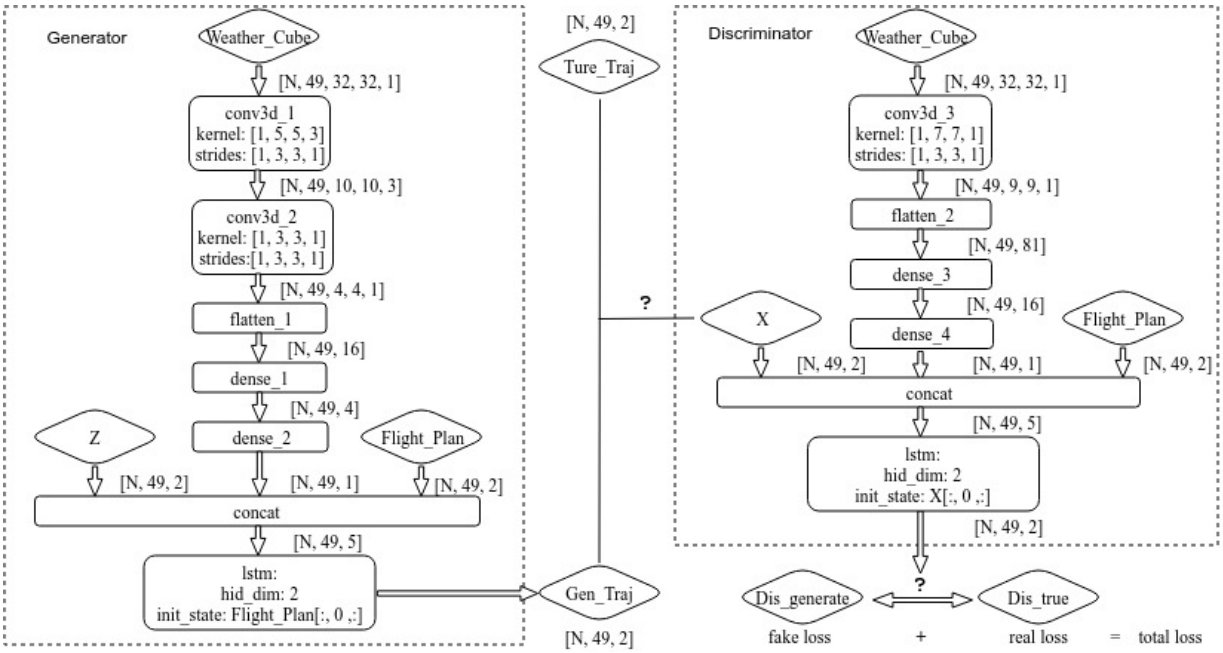

Fig. 5 CGAN Network Architecture

As discussed in the previous session, GAN architecture can be decomposed into two deep neural networks, the generator G and the discriminator D. A schematic show of our model can be seen from Fig. 5. In this plot, the inputs and outputs stay in the diamond boxes, N is the batch size used for training. The inputs to the discriminator are random noise tensor z, weather feature cube tensor w and the flight plan tensor p. The output of the generator is the generated trajectory. On the other side, the data feed into the discriminator are the conditional tensors (w and p) as well as the input x. When x is the ground truth data, then the output of the discriminator is used to calculate the true loss in the RMSE sense and if not, the output is used to compute the fake loss. The combination of real loss and the fake loss is the total loss of the network which is also called discriminator loss. The fake loss is called the generator loss. Then the problem is playing a mini-max game with objective functions as Eq. (4). The optimal solution is the best generator G^* that could output the predicted position of the accurately in Eq. (5) and G^* is also used for out-of-sample testing. We only interested in the optimal value of G^* is because we can verify if the generator works well but it's impossible to visualize the output from the discriminator. Different from other types of neural networks, we cannot determine whether the generative adversarial net is well trained or not by checking the loss value. The best approach is by visualizing the output of the generator. In the current problem, is to check the overall variance reduction as well as visualize the model prediction performance.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)}[log(D(x|w, p))] + \mathbb{E}_{x \sim P_z(z)}[log(1 - D(G(z|w, p)))] \tag{4}$$

$$G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) \tag{5}$$

Here are a few implementation details. The generator is formed by a convolutional neural network (CNN) and a simple LSTM layer. The conditions to the generator are the weather cube tensor and the flight plan tensor. The CNN includes two three dimensional convolutional layers and two fully-connect layers. It is used to extract weather features

from the weather tensors. The pooling operation between convolutional layers is omitted since the weather location makes an impact in our case. the kernel sizes for convolutional layers are $1 \times 5 \times 5 \times 3$ and $1 \times 3 \times 3 \times 1$, respectively. The last dimension of the kernel is also called the number of filters. The stride along the depth, height, and width of the weather tensor is $1 \times 3 \times 3 \times 1$ for these two layers. In our case, the depth is 49, the width and height are both 32. The first convolutional layer doesn't have padding while the second layer has a zero padding. After flatten along the last three dimensions, the output of convolutional layers is feed into two dense layers with a hidden dimension of 4 and 1. Dropout is used after each dense layer with a ratio of 0.5. Then the extracted weather features concatenate with the flight plan as well as the random noise tensor are fed into the LSTM layer. The hidden dimension of LSTM is set to be 2. The initial state of cell tensor and hidden tensor is set to be the first point of each data point.

The discriminator has a similar architecture as the generator network composes a CNN and a simple LSTM layer. However, the CNN only has one three dimensional convolutional layers in the discriminator. The kernel size of the layer is $1 \times 7 \times 7 \times 1$ and the stride is $1 \times 3 \times 3 \times 1$ with no padding operations. The following two full connect layers have a hidden dimension of 16 and 1. Dropout is used after each dense layer with a ratio of 0.5. Also, we concatenate the extracted weather tensor with the flight plan tensor to the input tensor x together first then feed it into a simple LSTM layer with a hidden dimension of 2. The initialization of LSTM is set to be the first point of input x.

The training process is performed on a workstation with Intel XEON E5-1620 v4 @3.50 GHz CPU and an NVIDIA GTX 1080 graphics card using tensorflow-gpu 1.6.0[‡]. The data is separated into a training set and a testing set with a weight of 0.8 and 0.2. Normalization is performed prior to training. The batch size is set to be 64 in consider of the card capacity. The training set is feed into the model during the training process and the testing set is used for the out-of-sample test. The best result comes from the trial with Adam optimizer with the learning rate of $1e^{-3}$ running 100 epochs. The backpropagation process of generator and discriminator are performed separately and with a steps ratio of 1:5.

V. Experimental Results

As we have described above, the testing result is the output from the generator with sampled random noise tensor z and condition tensors p and w as inputs. During each test, z is randomly sampled from the truncated normal distribution defined above. We perform 100 tests and calculate the mean values (μ_x, μ_y) and standard deviations (σ_x, σ_y) of the predicted coordinates from the results of these tests. The confidence range can be determined based on these values as confidence ellipses, in consideration of both the latitude and longitude coordinates are random output variables. Part of the results is visualized and shown in Fig. 6. The range of the plot is set as the range of the control sector ZID. Generally, we would like to make the red dots as close as possible to the green dots.

The result plots can be divided into two categories,

- The first case is there is a deviation between the flight trajectory and the flight plan as shown in Fig. 6 (a)(b)(c)(d)(f). It's clear to see from Fig. 6 (a)(b) that the uncertainty of the prediction increased where deviation happens. However, in Fig. 6 (a), the model is not able to predict the location of the last few points about arrival. This is because the landing procedure is different for each aircraft around the airport terminal. A purely data-driven approach for the landing procedure (or departing procedure) is not optimal and will cause trouble. Fig. 6 (b) shows that there is a large deviation but the model is not able to capture it. Same for Fig. 6 (c) where there is a deviation happened in the middle part of the sequence. Fig. 6 (d) has a similar behavior as Fig. 6 (a) near destination. The reason may indicate the model is not able to make a correct prediction but also may indicate the deviation is not caused by the convective weather conditions along the planned trajectory. Also, some of the predictions would still choose to stick with the flight plan (Fig. 6 (f)) when the deviation happens. This kind of phenomenon also happens in our previous work when using a straight-forward Conv-LSTM architecture.
- For the case that has no deviations between the flight plan and trajectory (Fig. 6 (e)), the mean prediction is acceptable but still shows uncertainties to the prediction. A possible reason for this is there are weather conditions at these coordinates but the aircraft choose to go through the convective weather region instead of trying to avoid it. Also, the pilot may choose vertical maneuvers in considering of fuel-saving and time delay issues.

We believe that pure visualization of each data point is not sufficient to measure the model performance. Statistical analysis is performed based on the definition of the research objectives using the mean value from performing many tests.

[‡]https://github.com/tensorflow/docs/tree/r1.6/site/en/api_docs

(a) Test Result #4

(b) Test Result #7

(c) Test Result #15

(d) Test Result #25

(e) Test Result #36

(f) Test Result #95

Fig. 6 Prediction Results Running 100 Tests

$$L2_k^{ori} = \sum_i^n \sum_j^d (Y_{k,i,j}^{true} - Y_{k,i,j}^{fp})^2 \tag{6}$$

$$L2_k^{new} = \sum_i^n \sum_j^d (Y_{k,i,j}^{true} - Y_{k,i,j}^{pred})^2 \tag{7}$$

$$reduction = \frac{Var(L2_k^{ori}) - Var(L2_k^{new})}{Var(L2_k^{ori})} \tag{8}$$

To better evaluate the model performance on deviation reduction, 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. (6) and Eq. (7). 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. (8) is the equation we used to calculate the overall variance reduction.

Table 1 Comparison to Previous Work

Models	Percentage of Flights Reduced	Overall Variance Reduction
Conv-LSTM	47.0%	12.3%
Generative Model	55.2%	22.1%

Table.1 shows the comparison of performance between the proposed generative approach and the previous Conv-LSTM approach. The generative model has a better prediction accuracy based on our evaluation metrics.

This proposed generative model achieves higher prediction accuracy and is able to produce a CI to the prediction. We also noticed that this model is still not doing the best as we acquired. The problems may come from multiple aspects. The data is fundamental to this research work. The data source we choose to use and the algorithms to take out the data play as the key role. The data should contain the pattern we would like the model to learn from. In our dataset, we have noticed that some of the data have this kind of pattern but a large part of the deviation has unknown reasons. These are the major difficulty of our research work. We would like to continue working towards a better performance in our future work.

VI. Conclusion

To address the issue of weather-related aircraft trajectory prediction task prior to takeoff, we purpose a CGAN architecture for trajectory calibration as well as uncertainty quantification. We define the generator as a combination of an LSTM layer and two convolutional layers without pooling operations to consider the spacial information. The discriminator is formed by another LSTM layer and a CNN as well. The optimization objective function is defined by minimizing the difference between the predicted tracks to the true tracks. The experiment is conducted using the flight data and weather data from NASA Sherlock Data Warehouse on the date of June 24th, 2019. The major benefits of this work are,

- The model uses a generative architecture that belongs to the semi-supervised learning task. This means we do not require a large amount of labeled data for training in considering the data limitation problem.
- The model can achieve better predicting performance compared to the previous Conv-LSTM model in the prediction population statistical study.
- The model is able to give a confidence interval to the prediction by doing many tests. The randomness comes from the randomly sampled input to the generator.

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. Kai Geobel and Dr. Anupa Bajwa, Principal

Investigator: Dr. Yongming Liu). The support is gratefully acknowledged. We also would like to thank Dr. Hao Yan from Arizona State University for helpful suggestions on the formulation of the model.

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] Erzberger, H., Lauderdale, T., and Chu, Y., "Automated conflict resolution, arrival management, and weather avoidance for air traffic management," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of aerospace engineering*, Vol. 226, No. 8, 2012, pp. 930–949.

[3] Swenson, H. N., Vincent, D., and Tobias, L., "Design and Operational Evaluation of the Traffic Management Advisor at the Ft. Worth Air Route Traffic Control Center," 1997.

[4] Davis, T. J., Isaacson, D. R., III, J. R., Den Braven, W., Lee, K., and Sanford, B., "Operational test results of the passive final approach spacing tool," *IFAC Proceedings Volumes*, Vol. 30, No. 8, 1997, pp. 175–181.

[5] Brudnicki, D., and McFarland, A., "User Request Evaluation Tool (URET) conflict probe performance and benefits assessment," *Proc. USA/Europe ATM Seminar*, Eurocontrol, 1997.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., 2014, pp. 2672–2680. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[11] Tran, L., Yin, X., and Liu, X., "Disentangled Representation Learning GAN for Pose-Invariant Face Recognition," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Dosovitskiy, A., Tobias Springenberg, J., and Brox, T., "Learning to generate chairs with convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1538–1546.

[13] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H., "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[14] 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.

[15] 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.

[16] Mirza, M., and Osindero, S., "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[17] Aggarwal, K., Kirchmeyer, M., Yadav, P., Keerthi, S. S., and Gallinari, P., "Regression with Conditional GAN," *CoRR*, Vol. abs/1905.12868, 2019. URL `http://arxiv.org/abs/1905.12868`.

[18] Yu, Y., and Canales, S., "Conditional LSTM-GAN for Melody Generation from Lyrics," *arXiv preprint arXiv:1908.05551*, 2019.

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

[20] Schmidhuber, J., "Learning factorial codes by predictability minimization," *Neural Computation*, Vol. 4, No. 6, 1992, pp. 863–879.

[21] Yu, L., Zhang, W., Wang, J., and Yu, Y., "Seqgan: Sequence generative adversarial nets with policy gradient," *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

[23] 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.

[24] LLC, O., "OpenNav Home Page," , 2018. URL `https://opennav.com/`.

[25] Klingle-Wilson, D., and Evans, J., "Description of the Corridor Integrated Weather System (CIWS) Weather Products," *Project Report ATC-317, MIT Lincoln Laboratory, Lexington, MA*, 2005.