Aircraft Trajectory Prediction using LSTM Neural Network with Embedded Convolutional Layer

Yutian Pang[1], Nan Xu[2], and Yongming Liu[3]

[1,2,3] *Arizona State University, Tempe, AZ, 85287, USA*
Yutian.Pang@asu.edu
nanxu2@asu.edu
Yongming.Liu@asu.edu

ABSTRACT

The development of convective weather avoidance algorithm is crucial for aviation operations and it is also a key objective of the next generation air traffic management system. This paper proposes a novel network architecture that embeds convolutional layers into long short-time memory (LSTM) cells to predict the trajectory, based on the convective weather condition along with the flight plan before the aircraft takeoff. The data used in the experiments are history flight track data, the last on-file flight plan, and the time-dependent convective weather map. The history flight data are taken from the NASA Sherlock database and the weather data used in this paper is the Echo Top (ET) convective weather product from Corridor Integrated Weather System (CIWS). The experiment is conducted using three months history data over the period from Nov 1, 2018 through Feb 5, 2019 with the flights from John F. Kennedy International Airport (JFK) to Los Angeles International Airport (LAX) but the methodology can be applied to the flights between any arbitrary two airports. Interpolation is performed on flight plans and real history tracks to fix the fold number of LSTM cells and also reduce computation complexity. The training loss is defined as the standard Mean Squared Error (MSE) of the predicted tracks and the real history tracks. Adam optimizer is used for backpropagation. Learning from the real historical flight data, the out-of-sample test shows that 47.0% of the predicted flight tracks can reduce the deviation compared to the last on-file flight plan. The overall variance is reduced by 12.3%.

1. INTRODUCTION

The design of future U.S. air traffic system, referred to as NextGen (H. Swenson, Barhydt, & Landis, 2006), is the subject of current research at universities and research centers around the country (Erzberger, Lauderdale, & Chu, 2012). While the developing of algorithms for convective weather avoidance contributes a lot to it since the weather-related delay of commercial operations is one of the most frequently encountered problems in the en-route airspace. It is able to reduce the risk of safety concerns. The convective weather conditions can develop rapidly and pose danger to mid-air traffic activities among the United States airspace. Once the convective weather condition is confirmed by the National Weather Service (NWS), the relevant information would be sent to the controller and the pilot to update the flight route to avoid flying through the regions without creating conflicts with other aircraft with the surrounding traffic. This kind of activity will largely increase the workload of controllers, especially when a large number of airplanes are heading to the same region. In NextGen, information sharing between aircraft will be greatly enhanced so that each aircraft receives and transmits the cooperative surveillance information, and thus aircraft can take over a certain amount of ATM tasks from ground air traffic controllers. These tasks include flight plan changes, trajectory prediction, and conflict detection. In this way, NextGen can alleviate the workloads of ground air traffic controllers (W. Liu & Hwang, 2011; Phillips, 1996; Paielli & Erzberger, 1997). A study with Dallas/Fort Worth ARTCC airspace shows that the controller's workload is highly related to dynamic density. It also shows that cognitive workload aspects are important because past research indicates that infrequent but critical events such as loss of separation, altitude deviations, and incorrect pilot read back impose considerable mental workload on the controllers (Sridhar, Sheth, & Grabbe, 1998). Thus the development of an automated trajectory planning tool is crucial to aviation operations within United States Airspace to address the issues caused by the increasing workload of controllers/pilots. The aircraft weather avoidance algorithm can be divided into the following two categories,

Dynamic Weather Reroutes (DWR) (McNally et al., 2012) is raised over the past several years as a ground-based concept to efficiently and automatically generate trajectories around convective weather regions for in-flight aircraft within the enroute airspace. It is a search engine that continuously and automatically analyzes in-flight aircraft in the airspace and proposes a time and fuel-saving corrections to current weather avoidance routes (McNally et al., 2012). The system Terminal AutoResolver (Erzberger, Nikoleris, Paielli, & Chu, 2016) was developed to employs a variety of maneuvers including resolve conflicts and avoid weather cells with the algorithm to find the path around the convective weather polygon using a single auxiliary waypoint (Erzberger et al., 2012). NASA and American Airlines (AA) have been conducting a trial of DWR at AA's Integrated Operations Center in Fort Worth, Texas since July 2012 using DWR. Results show that AA flights with DWR in use realize about 20% more savings than non-AA flights. A weather forecast analysis examines the extent to which DWR routes rated acceptable by AA users remain clear of downstream weather. A sector congestion analysis indicates congestion could be reduced 19-38% if all flights fly DWR routes rather than nominal weather avoidance routes (McNally et al., 2013, 2015).

Strategic Trajectory Prediction (TP) in the both spatial and temporal domain is another popular research topic in society. The use of ground-based TP tools has been applied to air traffic controllers with traffic management (H. N. Swenson, Vincent, & Tobias, 1997), efficient runway utilization (Davis et al., 1997), conflict detection (Brudnicki & McFarland, 1997) and also on weather avoidance (Ayhan & Samet, 2016). 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 (Lymperopoulos, Lygeros, & Lecchini, 2006). Similarly, another recent research study applies the Hidden Markov Model (HMM) to predict trajectories taking environmental uncertainties into account (Ayhan & Samet, 2016). 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 and require the dataset to cover all possible past flight routes for trajectory selection. In contrast, a few researchers emphasize on probabilistic approach rather than deterministic prediction. 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 (De Leege, van Paassen, & Mulder, 2013). 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 network structure for this task (Y. Liu & 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 is able to 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.

Long short-term memory (LSTM) neural network has been shown as an effective tool in multiple sequential learning tasks compared to a fully connected neural network (FCN) and convolutional neural network (CNN). LSTM is able to handle long-range time series data, noisy data, and continuous values. Furthermore, it doesn't require a *priori* choice of a finite number of state variables like HMM (Hochreiter & Schmidhuber, 1997). LSTM has shown to be a powerful tool in natural language translation (Cho et al., 2014; Venugopalan et al., 2014; Sutskever, Vinyals, & Le, 2014), speech recognition (Graves & Jaitly, 2014; J. Chorowski, Bahdanau, Cho, & Bengio, 2014). Another interesting research work focuses on predicting the runway incursion during the landing process of the aircraft (Wang, Pang, Liu, Dutta, & Yang, 2019). The purposed Bayesian updating model is able to predict the occurrence of runway incursion using the flight track data. By incorporating physics into the network, a novel hybrid learning model is proposed to predict the flight trajectories faster and more reliably (Yu, Yao, & Liu, 2019, 2018). Another novel application is "Social LSTM" (Alahi et al., 2016), which predicts the human trajectory in crowded areas. By modeling each individual with an LSTM network and using pooling techniques to connect each LSTM networks, Social LSTM can achieve a far better prediction accuracy than other works. A recent study on aircraft trajectory prediction incorporates aircraft dynamics with recurrent neural network and purposed a deep residual RNN architecture called physics-based learning (Yu et al., 2019, 2018). This special form of RNN structure can accurately predict aircraft responses and shows excellent extrapolation performances. It's also worth pointing out that the recurrent neural network has difficulties in the backpropagation process in training. The concept of gradient vanishing and gradient explosion (Bengio, Simard, Frasconi, et al., 1994) was raised up by researchers during applications.

Inspired by the research of 4D trajectory prediction (Y. Liu & Hansen, 2018) and the characteristics of RNN, we apply a special form of the recurrent neural network, LSTM in particularly, to aircraft trajectory prediction task. While the traditional fully connected LSTM (FC-LSTM) doesn't take spatial correlation into consideration, we embed two convolutional layers into LSTM cells thus increase the size of hidden tensors to cooperate with weather information into each of the LSTM gates. This also makes the LSTM "wider" than FC-LSTM with more parameters for the model to update in the backpropagation process.

This paper is organized as follows: Section 1 briefly introduces the background and current research progress in weather-related air traffic management field. This includes discussion on the two main branches in this area, Dynamic Weather Reroutes (DWR) and Strategic Trajectory Prediction (TP), followed by a concise review of LSTM and how we define the problem with our model. Section 2 talks about the data acquisition and processing of FAA flight data and CIWS convective weather products. In Section 3, we explain our model architecture in detail. Section 4 shows the training and the out-of-sample testing result of the purposed model architecture.

2. DATA ACQUISITION

All data used in this paper are obtained from the Sherlock Data Warehouse (Arneson, 2018). Sherlock is a big data system for data visualization to support air traffic management (ATM) research, which includes a database, a web-based graphical user interface (GUI) 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 (Arneson, 2018). Data of Sherlock comes primarily from the FAA and the National Oceanic Atmospheric Administration (NOAA) (Eshow, Lui, & Ranjan, 2014). There are multiple sources feeding into Sherlock, such as flight plan and track from Air Route Traffic Control Center and TRACON, 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 SWIM data sources for flight data. The raw data Sherlock collected will be parsed, processed and stored on an Oracle server and the approved user is able to choose the date and data source in order to query the database and download. Here we only use the data from two sources of them. The Integrated Flight Format (IFF) flight data and Echo Top (ET) convective weather data from CIWS.

2.1. Flight Data

The ATAC data is collected from 76 different FAA facilities in the Integrated Flight Format (IFF). The IFF data format includes all source data plus derived fields about flight summary, track points, and flight plan,

> The flight summary information includes flight time, flight call sign, aircraft type, origin and destination source, and the flight operation type. It is a general description of the flight.
>
> The flight track points are the record of real flight operation. It includes the total 4D flight tracks, ground speed, and climb rate. It also contains information about the departing, cruising and landing procedure.
>
> The flight plan comes as a string of waypoints. We use the online aviation database OpenNav (LLC, 2018) to parse the string of flight plan into WGS84 coordinates.

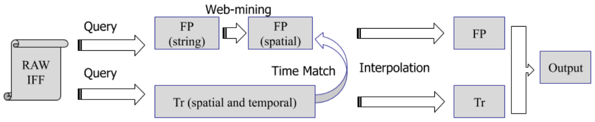

Figure 1. Process Raw IFF data

The raw IFF data download from Sherlock is massive with more than half-million rows for one day which contains all the flight plans, flight tracks and other flight information of aircraft flying above the United States airspace within one day. That is 30k-50k flight call signs for each day's file recorded. Thus, multiple data parsers and filters are created for raw data cleaning. Fig. 1 graphically shows the workflow for preprocessing the IFF data. Fig. 2 is a plot of flight tracks from JFK to LAX on Nov 1, 2018, taken from the IFF data. Each flight has a unique flight trajectory even if the controller tends to make the flight plan for flight call signs from the same operator are identical.

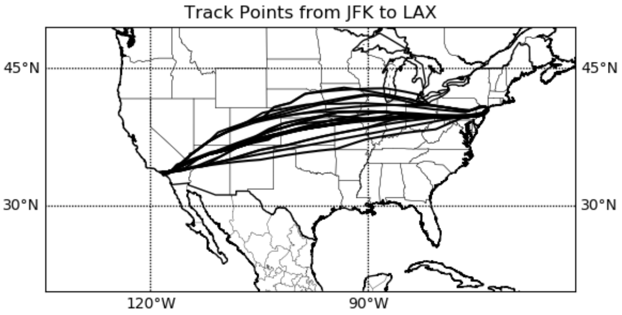

Figure 2. Flight tracks from JFK to LAX on 11/01/2018

Alg.1 is a detailed explanation of the flight data processing part. The inputs are the departing airport code Key1 and arriving airport code Key2, for instance, JFK and LAX. DateList is the list of dates for the raw data. We downloaded three months' data over the period from Nov 1, 2018, through Feb 5, 2019, of approximately 100 days' data thus the DateList contains these 100 days. The last input should be the number of points in the output tracks and flight plans, this is also the folder numbers in LSTM. The output is the processed true trajectory Tr_{new} and flight plan FP_{new} of a fixed equal length of n for each flight call sign. It worth pointing out that the idea of a given flight plan coordinates a time-stamp comes from the requirement of point matching of each predicted trajectory points and actual trajectory points. Although the actual aircraft will not flying at a constant ground speed during takeoff and landing procedures, we still choose to use linear interpolation to FPs and flight tracks to simplify the data processing part.

Algorithm 1 Flight Tracks and Flight Plans

```
Data: IFF raw data path
Input : Key1, Key2, DateList, n
Output: $Tr_{new}$, $FP_{new}$
for date in DateList do
    Load IFF raw data file of given date
    Query the file departing at Key1 and arriving at Key2
    Create a folder named with date
    Save 4D Tr and last on-file FP named with flight call sign and date in the datefolder
    for callsign in datefolder do
        Parse FP into WGS84 coordinates
        Match FP from Tr for the time column
        Interpolate FP and Tr with 1 second interval
        Equally sample n points from FP and Tr as $Tr_{new}$ and $FP_{new}$
        Save $Tr_{new}$ and $FP_{new}$
    end for
end for
```

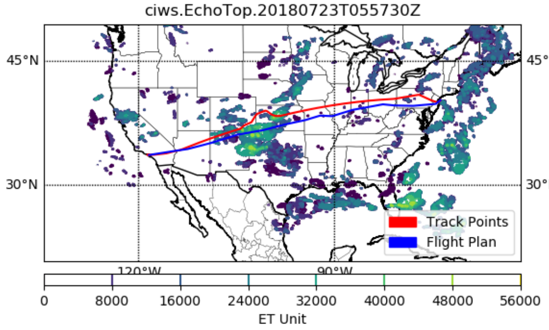

Figure 3. EchoTop contour for JBU1823 on 7/23/2018

Table 1. EchoTop Key Features

Parameters	Current	Forecast
Dimension	1x1x3520x5120	24x1x3520x5120
Range Latitude / °	[19.36, 48.90]	
Range Longitude / °	[-134.35, -61.65]	
Update Frequency / s	150	300

2.2. Weather Data

The weather data are fetched from Corridor Integrated Weather Systems (CIWS). CIWS is designed to improve convective weather decision support for congested en-route airspace (and the terminals that lie under that airspace) by automatically generating graphical depictions of the current severe weather situation and providing frequently updated forecasts of the future weather locations for forecast times from zero to two hours. 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 (Klingle-Wilson & Evans, 2005). The two key features of CIWS, Echo Top (ET) and Vertically Integrated Liquid (VIL), both come with current and forecast datasets in Sherlock. A Lincoln Laboratory's study shows that vertically integrated liquid (VIL) is a better indicator of storm severity and new growth and is less susceptible than other precipitation representations to anomalous propagation and other anomalies (Robinson, Evans, & Crowe, 2002).

While the echo top weather product is another feature to measure the severe weather conditions such as thunderstorms. Another study of Lincoln Laboratory proposes an improved echo tops algorithm that could more accurately estimate the true storm echo tops (Evans & Ducot, 2006; Evans, Carusone, Wolfson, Crowe, & Smalley, n.d.). The current weather dataset is updated every 150 seconds and the forecast weather dataset is updated every 300 seconds. Figure. 3 is a schematic plot of ET at 5:57:30 AM on July 23, 2018, within the NAS. The real history tracks not following the flight plan but deviate to fly through the gap of convective weather contours. This also shows the importance of aircraft trajectory adjustments under convective weather influence. A few other key parameters of ET data are shown in Table. 1.

Algorithm 2 Weather Cube Generation

```
Data: True Trajectory Data, Weather Data
Input : TrajectoryPoints, CubeSize, Weather
Output: $WeatherCube_{coord}$, $WeatherCube_{Value}$
Convert weather data coordinates to Mercator's Projection Coordinates
Convert trajectoryPoints coordinate to Mercator's Projection Coordinates
for Points in TrajectoryPoint do
    Determine the flight direction
    Find a line perpendicular to the current position normal line
    Generate 20 points along this perpendicular line
    for Step in the CubeSize do
        From every right of these 20 points move to the next point which passed through a perpendicular line parallel to the previous line with distance of Step
        Generate 20 points along the new perpendicular line
        Move another step from very right of these new 20 points
    end for
end for
Save $WeatherCube_{coord}$ and $WeatherCube_{Value}$
```

In this research, we only use the current ET data as it's more holistic than the forecast data. Nevertheless, it can apply to forecast dataset without too much effort. Since the key idea of this research is predicting real trajectories using the flight plan and convective weather close to the flight plan. We developed Alg.2 to take out the convective weather cubes ahead of the current position of the aircraft. The size of the weather cubes is $20 \quad 20$ after scaling the original resolution by 10 times and the angle of the cube is set as the aircraft heading angle. However, the cube size can be changed as of the user's interest. This means the convective weather area of interest can be expanded or shrank under different settings. A schematic plot of the taken weather cubes is shown in Fig. 4. The blue area is the range of convective weather we are going to take out for the specific flight plan. The width of the blue band can be changed by the cube size parameter in Alg.2. The black box shows the exactly feature cube area at the current position. Theoretically, there is one box region for each trajectory point.

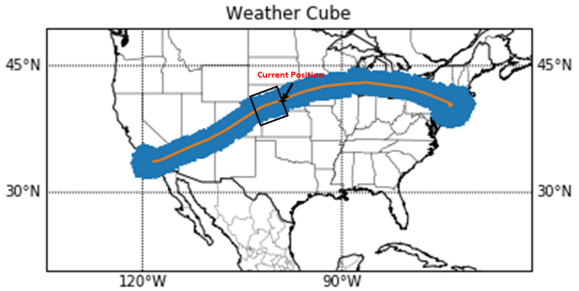

Figure 4. Schematic of Weather Cube Algorithm

We have collected the data over the range from Nov 1, 2018, to Feb 5, 2019, of 2737 flight tracks and flight plans. While only 2528 of them have complete convective weather data during the period from departure to arrival. Only longitude and latitude coordinates are considered in this experiment.

3. NETWORK ARCHITECTURE

Aircraft tend to frequently deviate from the last on-file flight plan due to weather-related issues discussed above. We formulated the trajectory prediction task as a sequence learning problem of calibrating the flight plan to be as close as possible with the actual flight trajectory using convective weather features. The model has been defined as a *point matching* regression model between the FP and Tr. The inputs to the model would be the flight plan and the weather cubes ahead of each point of the flight plan and the output is the predicted trajectory. LSTM has been proved efficient to handle sequential learning problem while the convolutional layers are introduced into the architecture to extract useful features from weather cube. The outputs of convolutional layers are concatenated with LSTM hidden tensors thus feed into the calculation within LSTM cells.

3.1. LSTM

LSTM neural network, as a special form of RNN, was first introduced in 1997 (Hochreiter & Schmidhuber, 1997). All recurrent neural networks have a repeating module of the neural network, for instance, a single *Tanh* layer in a standard RNN. Each of the repeating modules is called an RNN fold and the fold number is just the number of repeating modules. LSTM also has this chain-like structure with different repeating modules. A LSTM cell consist of three cell gates, the forget gate (Eq. 1b), input gate (Eq. 1c) and output gate (Eq. 1f). These three gates are connected by the cell state tensor in Eq. 1d and Eq. 1e. The forget gate is used to decide whether the information needs to be pass into the cell state or not. The *Sigmoid* layer here output a number between 0 and 1. A 1 usually means the information needs to be thrown away. The input gate is to control whether the information is going to store in the cell state thus another "yes or no" operation to the cell state is performed after the input gate (Eq. 1d). The *Tanh* layer here creates new candidate values $\hat{C}_t$ and the cell state is updated with the output of the input gate and the new candidate (Eq. 1e). At last, we are going to decide what we are going to output in the output gate (Eq. 1f and Eq. 1g). *Tanh* will push the cell state through it to a value between -1 and 1 and multiply it to the output of the input gate which is a result after sigmoid operation. Thus LSTM will only output what we would like to output. The symbol means concatenate operation between arrays.

$$h_x = h_t \quad x_t \tag{1a}$$

$$f_t = Sigmoid(W_f \quad h_x + b_f) \tag{1b}$$

$$i_t = Sigmoid(W_i \quad h_x + b_i) \tag{1c}$$

$$\hat{C}_t = Tanh(W_c \quad h_x + b_c) \tag{1d}$$

$$c_t = f_t \quad c_t + i_t \quad \hat{C}_t \tag{1e}$$

$$o_t = Sigmoid(W_o \quad h_x + b_o) \tag{1f}$$

$$h_t = o_t \quad Tanh(c_t) \tag{1g}$$

LSTM is capable of performing two kinds of sequential learning tasks, sequence to sequence (Seq2Seq) learning and time series forecasting. Seq2Seq learning is a widely adopted method in machine translation and question and answering (QA) systems. It was specially designed to learn and predict 1D sequences, such as complete sentences in text or speech (Karatzoglou, Jablonski, & Beigl, 2018). It has been shown successful in machine translation (Chung, Gulcehre, Cho, & Bengio, 2014; Luong, Sutskever, Le, Vinyals, & Zaremba, 2014; J. K. Chorowski, Bahdanau, Serdyuk, Cho, & Bengio, 2015), and in other natural language processing (NLP) tasks such as parsing (Cheng, Dong, & Lapata, 2016), text summarizing (Nallapati, Zhai, & Zhou, 2017) and multi-task learning (Luong & Manning, 2016). The work performed in this paper can be classified as a Seq2Seq learning approach.

One of the key ideas behind LSTM is that it can connect the previous information to the current state. This is achieved by the high dimensional array called hidden tensor and the cell state across all the calculation of three gates. The information is extracted in each fold of LSTM and is stored by the hidden tensor generated from the current state. Then the hidden tensor will be passed into the next LSTM fold thus the information from the previous state is introduced into the calculation of each gate. The short-term memory thus can be remembered by the model fold by fold thus formulate a long period sequential learning model.

3.2. Architecture

Compared to the classical architecture of LSTM, we modify the repeating module of each LSTM folds to incorporate with the convective weather cubes. We add two convolutional layers and two dense layers before the computing of forget gate (Eq. 2a, 2b, 2c, 2d). These layers are used to extract useful weather features into a vector and concatenate them with the hidden tensor of LSTM to pass through iterations. To increase the number of parameters included in the LSTM cells, we also increase the dimension of hidden tensors by adding a fully-connect (FC) layer (Eq. 2l). This will also increase the number of parameters in each gate computations drastically. Another graphically drawing of the unfolded network architecture is shown in Fig. 5. The input to each LSTM fold would be the current position $(x_i; y_i)$ and the convective weather cube ahead of the current position $Cube_i$, the output from each fold would be the predicted position $(X_i; Y_i)$ under the calibration of weather cubes. Then the final predicted trajectory is just the stack of each calibrated position.

$$x_{conv1} = Relu(K_1 \quad x_{weather}) \tag{2a}$$

$$x_{conv2} = Relu(K_2 \quad x_{conv1}) \tag{2b}$$

$$x_{dense1} = Relu(Dense(flatten(x_{conv2}); 16)) \tag{2c}$$

$$x_{dense2} = Relu(Dense(x_{dense1}; 4)) \tag{2d}$$

$$h_x = h_t \quad x_t \quad x_{dense2} \tag{2e}$$

$$f_t = Sigmoid(W_f \quad h_x + b_f) \tag{2f}$$

$$i_t = Sigmoid(W_i \quad h_x + b_i) \tag{2g}$$

$$\hat{c}_t = Tanh(W_c \quad h_x + b_c) \tag{2h}$$

$$c_t = f_t \quad c_t + i_t \quad \hat{c}_t \tag{2i}$$

$$o_t = Sigmoid(W_o \quad h_x + b_o) \tag{2j}$$

$$h_t = o_t \quad Tanh(c_t) \tag{2k}$$

$$h_t = Relu(Dense(dim_{h_t}; dim_{hidden} \quad 6)) \tag{2l}$$

Here are a few implementation details,

The data are feed into the model after min-max normalization which linearly maps the data into the range between 0 and 1.

The hidden dimension can choose any arbitrary integer but is set to be 100 in our case.

The hidden tensor and cell state tensor is initialized with the start point of the trajectory which normally is the departing airport coordinates. All the other parameters such as weight and bias tensor are initialized with a normal distribution of 0 mean and standard deviation of 0.1 for convenience.

The first convolutional layer uses a kernel of size 6 6 2 and the second convolutional layers has a kernel of size 3 3 4, both with a stride of 2.

There are no pooling operations between convolutional layers since the location of weather cells is vital to the output.

It's not easy to modify the layer functions of the high-level deep learning packages such as Keras thus the work is performed under the Tensorflow environment.

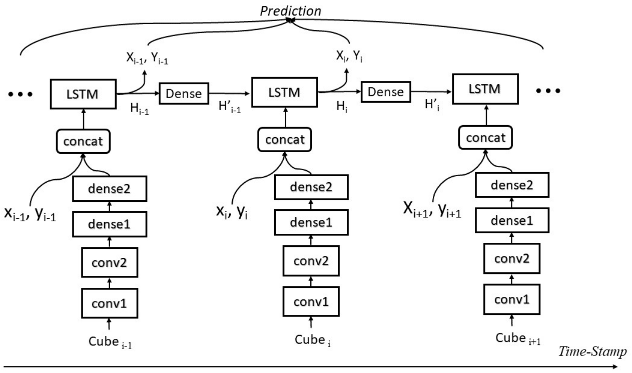

Figure 5. Unfolded Network Architecture

The loss function is defined as the mean squared error (MSE) between the predicted coordinates and flown trajectory as shown in Eq. 3.

$$L(W_x; b_x) = \frac{1}{n} \sum_i (Y_i^{pred} \quad Y_i^{true})^2 \tag{3}$$

4. RESULTS

The training process is performed on a workstation with Intel Xeon E5-1620 v4 @3.50 GHz chipset and an Nvidia GTX 1080 graphics card. Running 1000 epochs with a batch size of 64, the MSE reaches $1 \quad 10^{\quad 3}$. The parameters such as hidden dimensions, convolutional layers' kernel sizes, number of hidden units and even the choice of the optimizer contribute a lot to successfully give back an acceptable output. While it worth pointing out that 90% of the data is split as the training set and 10% of the data is split as the testing set. One needs to continuously adjust the model parameter settings during the training process.

The final input of the model is the parsed, interpolated flight plan route including horizontal coordinates and the corresponding weather feature cube at each point and the outcome of the model is the predicted trajectory of each flight call sign. After inverse normalization of the predicted output of the model, we draw out the model output as the calibrate predicted trajectory, the original last on-file flight plan, and the ground truth real history flew flight tracks along with the weather conditions together as shown in Fig. 6. The plot visualizes the flight VRD 415 on Apr. 7, 2017 where the relevant data are taken from the testing data set. We could see that the prediction tracks are closer to the aircraft true tracks. We also notice that the prediction tends to stay close to the original flight plan.

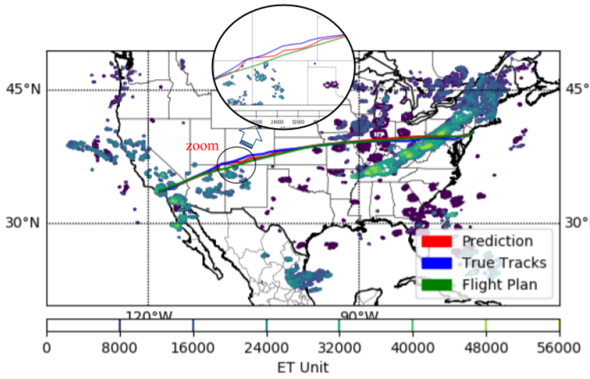

Figure 6. A Plot of Testing Result

To better illustrate the prediction improvement of the neural network, we compared the result from the original deviation and the calibrated deviation. The original deviation is defined as the l-2 norm between the ground truth history flight tracks and the original flight plan route while the calibrated deviation is the l-2 norm between the ground truth history flight tracks and the predicted flight tracks as shown in Eq. 4 and Eq. 5. The n here stands for the LSTM fold numbers and 2 is a representation of longitude and latitude.

$$L2_k^{ori} = \sum_i \sum_j (Y_{k;i;j}^{true} - Y_{k;i;j}^{fp})^2 \tag{4}$$

$$L2_k^{new} = \sum_i \sum_j (Y_{k;i;j}^{true} - Y_{k;i;j}^{pred})^2 \tag{5}$$

Using this evaluation metric, each flight call sign has a pair of the original l-2 norm and new l-2 norm values. We could easily get the percentage of deviation reduced in the testing data set by count how many $L2_k^{ori}$ is bigger than $L2_k^{new}$ out of the total number of the testing dataset. Index k here is the index for data set where one flight track as one data set. It's shown that 47.0% of the deviation between flight tracks are reduced in our trained model. Then we calculated the variance improvement of deviations using Eq. 6. The result shows that the overall l2-norm (flight track deviation) reduction among the testing set is 12.3%.

$$reduction = \frac{Var(L2_k^{ori}) - Var(L2_k^{new})}{Var(L2_k^{ori})} \tag{6}$$

5. CONCLUSION

This paper aims at addressing the issue of convective weather-related aircraft trajectory prediction prior to takeoff. Inspired by the research of 4D trajectory prediction (Y. Liu & Hansen, 2018) and the characteristics of RNN, we apply a special form of the recurrent neural network, LSTM in particularly, to aircraft trajectory prediction task. While the traditional fully connected LSTM (FC-LSTM) doesn't take spatial correlation into consideration, we embed two convolutional layers into LSTM cells thus increase the size of hidden tensors to cooperate with weather information into each of the LSTM gates. This also makes the LSTM "wider" than FC-LSTM with more parameters for the model to update in the backpropagation process. The developed neural network model has two major differences compared to the classical FC-LSTM network and will be discussed later.

As part of the NASA University Leadership Initiative (ULI) project (Y. Liu & Goebel, 2018) which aims to address the safety needs and their technology solutions for future national airspace system (NAS), we developed a novel convolutional layers embedded LSTM neural network 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.

To reach this objective of the research, we have the following major innovations,

1. Generate 4D flight plan from text format string flight plans using web-mining tools and real history flight trajectory data as a reference.
2. Develop an algorithm to take out the convective weather feature cubes of given coordinates and convective weather data. The size of the cube is a user-defined parameter which means the weather area of interest can be changed as of user's interest. The angle of the feature cubes is rotated with the flight heading angle.
3. Embed convolutional layers into the repeating modules of the LSTM structure to take our useful information inside of the convective weather feature cubes. Omit the use of pooling layers to keep the location information.
4. Expand the hidden tensor dimensions by adding additional fully connected layers within the LSTM folds to increase the number of parameters of the network.

The training process shows that after training for 1000 epochs, the model's mean squared error is able to reach $1 \quad 10^{3}$. The dimension of hidden tensors in LSTM fold and the number of hidden units is decided during training. The final model outcome shows that our model prediction can calibrate the last on-file flight plan with the given convective weather feature cubes. Statistical study shows that 47.0% of the flight deviation is reduced using our model and the overall variance of deviation reduction is 12.3%. To output a calibrated trajectory, the model will need a flight plan and convective weather information as input to generate a calibrated flight trajectory. The uniqueness of the purposed NN architecture comes from the modification of LSTM recurrence. Instead of simply layers stacking, we include convolutional operations into the flow of recurrence.

ACKNOWLEDGMENT

All 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.

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
AA	american airlines
HMM	hidden markov model
GLM	generalized linear model
FCN	fully connected network
CNN	convolutional neural network
$NOAA$	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
FC	fully connected
MSE	mean squared error
	concatenate operation symbol
	convolution operation symbol

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).

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

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* (pp. 21–30).

Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166.

Brudnicki, D., & McFarland, A. (1997). User request evaluation tool (uret) conflict probe performance and benefits assessment. In *Proc. usa/europe atm seminar.*

Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chorowski, J., Bahdanau, D., Cho, K., & Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems* (pp. 577–585).

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

Davis, T. J., Isaacson, D. R., III, J. R., Den Braven, W., Lee, K., & Sanford, B. (1997). Operational test results of the passive final approach spacing tool. *IFAC Proceedings Volumes*, *30*(8), 175–181.

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).

Erzberger, H., Lauderdale, T., & Chu, Y. (2012). 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*, *226*(8), 930–949.

Erzberger, H., Nikoleris, T., Paielli, R. A., & Chu, Y.-C. (2016). Algorithms for control of arrival and departure traffic in terminal airspace. *Proceedings of the In-*

stitution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, *230*(9), 1762–1779.

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)* (pp. 1E3–1).

Evans, J. E., Carusone, K., Wolfson, M. M., Crowe, B., & Smalley, D. J. (n.d.). 10.1 multi-radar integration to improve en route aviation operations in severe convective weather.

Evans, J. E., & Ducot, E. R. (2006). Corridor integrated weather system. *Lincoln Laboratory Journal*, *16*(1), 59.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (pp. 1764–1772).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Karatzoglou, A., Jablonski, A., & Beigl, M. (2018). A seq2seq learning approach for modeling semantic trajectories and predicting the next location. In *Proceedings of the 26th acm sigspatial international conference on advances in geographic information systems* (pp. 528–531).

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*.

Liu, W., & Hwang, I. (2011). Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics*, *34*(6), 1779–1789.

Liu, Y., & Goebel, K. (2018). Information fusion for national airspace system prognostics. In *Phm society conference* (Vol. 10).

Liu, Y., & Hansen, M. (2018). Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. *arXiv preprint arXiv:1812.11670*.

LLC, O. (2018). *OpenNav home page.* Retrieved 2019-04-30, from `https://opennav.com/`

Luong, M.-T., & Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2014). Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

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).

McNally, D., Sheth, K., Gong, C., Borchers, P., Osborne, J., Keany, D., ... others (2013). Operational evaluatioin of dynamic weather routes at american airlines.

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).

McNally, D., Sheth, K., Gong, C., Sterenchuk, M., Sahlman, S., Hinton, S., ... Shih, F.-T. (2015). Dynamic weather routes: two years of operational testing at american airlines. *Air Traffic Control Quarterly*, *23*(1), 55–81.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first aaai conference on artificial intelligence.*

Paielli, R. A., & Erzberger, H. (1997). Conflict probability estimation for free flight. *Journal of Guidance, Control, and Dynamics*, *20*(3), 588–596.

Phillips, E. H. (1996). Free flight poses multiple challenges. *Aviation Week & Space Technology*.

Robinson, M., Evans, J., & Crowe, B. (2002). En route weather depiction benefits of the nexrad vertically integrated liquid water product utilized by the corridor integrated weather system. In *10th conference on aviation, range and aerospace meteorology, american meteorological society, portland, or.*

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).

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).

Swenson, H., Barhydt, R., & Landis, M. (2006). *Next generation air transportation system (ngats) air traffic management (atm)-airspace project* (Tech. Rep.). Technical report, National Aeronautics and Space Administration.

Swenson, H. N., Vincent, D., & Tobias, L. (1997). Design and operational evaluation of the traffic management advisor at the ft. worth air route traffic control center.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., & Saenko, K. (2014). Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.

Wang, Y., Pang, Y., Liu, Y., Dutta, P., & Yang, B.-J. (2019). Aircraft trajectory prediction and risk assessment using bayesian updating. In *Aiaa aviation 2019 forum* (p. 2936).

Yu, Y., Yao, H., & Liu, Y. (2018). Physics-based learning for aircraft dynamics simulation. In *Phm society conference* (Vol. 10).

Yu, Y., Yao, H., & Liu, Y. (2019). Aircraft dynamics simulation using a novel physics-based learning method.