Lab Report

Title: (Final Report) Air Traffic Analysis
Notice: Dr. Bryan Runck
Author: Megan Marsolek
Date: 12/21/21

Project Repository: https://github.com/mmarsole/GIS5571FinalProject
Google Drive Link: NA
Time Spent: 16 hours

Abstract

Air Travel has become an increasing and popular means of travel, with airlines expanding and offering more flights to more destinations, it becomes a challenge to monitor and safely manage airspace to avoid collisions around highly frequented areas like airports. There are many regulations in place to assure travel safety, but it has the potential to benefit from incorporating Machine Leaning (ML) to better estimate a plane's near future location based on constantly changing variables (weather) and fixed restrictions (airspace classifications).

Using a plane's current and past locations (as well as it's heading and velocity) I intend to track and predict a plane's location in space over time (lat., lon.). Relying on Automatic Dependent Surveillance – Broadcast (ADS-B) for data, I will convert and prep the data for a ML model, and then train and test the accuracy by comparing it to the recorded flight path. In the end I will visually display a sample of the ML predicted flight path with that of the real flight path.

Problem Statement

The intention is to predict a plane's flight path based on data that may influence a plane's course. Given I have access to its present and past locations as it flies (via ADS-B data) I aim to predict where it will likely be in 2D space for some specified time in the future (seconds to minutes).

Table 1. Relevant (attainable) variables for the problem

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	Aircraft locations	Point data (Lat., Long., Alt.) from ADS-B (currently in csv file)	Point data in csv (extracted from TAR files)	lat, lon, geoaltitude, heading, velocity, time	ADS-B Data	Subset the data to isolate a couple of flight paths (for instance flights that travel through Chicago Class B airspace). Convert lat lon from excel to point data in Arc.
2	Airspace Classification B	Regulated airspace across the US with flying permissions that dictate flight protocol (subset to focus only on class B airspace)	2D shapefile (vector) or access to a 3D Google kmz file	UPPER_VAL, LOWER_VAL, NAME, SECTOR, LEVEL	FAA Airspace Shapefiles	Matching CRS with all other data, Subsetting data to just class B

Input Data

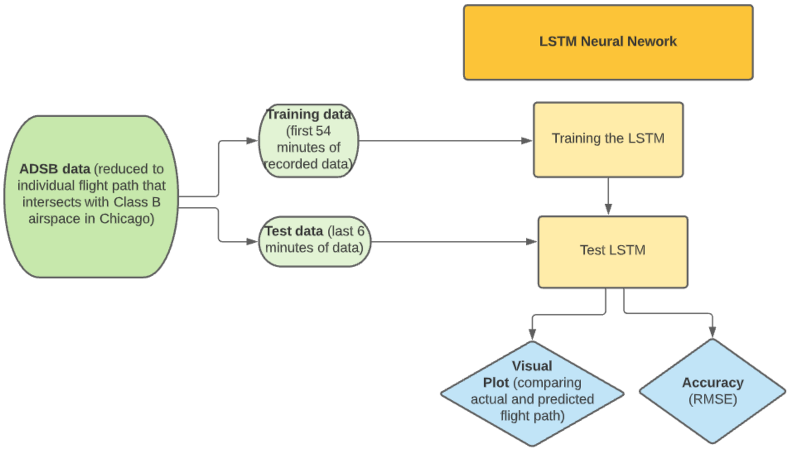
This project’s initial steps focused on subsetting and assessing the ADS-B flight data, which is a snapshot of all global flights between 12:00:10 AM and 12:59:50 AM on May 25th of 2020 (recorded in GMT time).

For simplicity, the flight data was subsetted to several smaller datasets, first by reducing the data to only contain flight paths that intersected with Chicago’s Class B airspace (this airspace surrounds O’Hare International Airport and Chicago Midway International Airport). This step reduced the csv file from its original 723,098 points (over 675,000 flight paths) to 11,697 points (57 flight paths), a more manageable dataset size. Then I subsetted the data to individual flight paths (based on the icao24 identifier flight number). These individual flight paths contained between 150 points (also referred to as timesteps, with increments as frequent as every 5 seconds) to 450 points. A sample of isolated flight paths were then used to train ML models in addition to one model being trained on all 57 flight paths intersecting the Chicago airspace.

Table 2. Data Access

#	Title	Purpose in Analysis	Link to Source
1	ADS-B	ADS-B data records the location of a plane at a given time in points (lat., long. Alt.). Its location can help predict where it might be at some time in the future. It also contains other attributes: make/model of aircraft, time, flight number, etc.	Open Sky Network
2	Class B,C,D,E Airspace Shape Files	Define where Airspace classifications are in 3D space that may affect flight paths (represented as 2D but contains attributes that describe its min and max altitudes).	Federal Aviation Administration (FAA)

Methods


LSTM Neural Nework
ADSB data (reduced to individual flight path that intersects with Class B airspace in Chicago)
Training data (first 54 minutes of recorded data)
Test data (last 6 minutes of data)
Training the LSTM
Test LSTM
Visual Plot (comparing actual and predicted flight path)
Accuracy (RMSE)

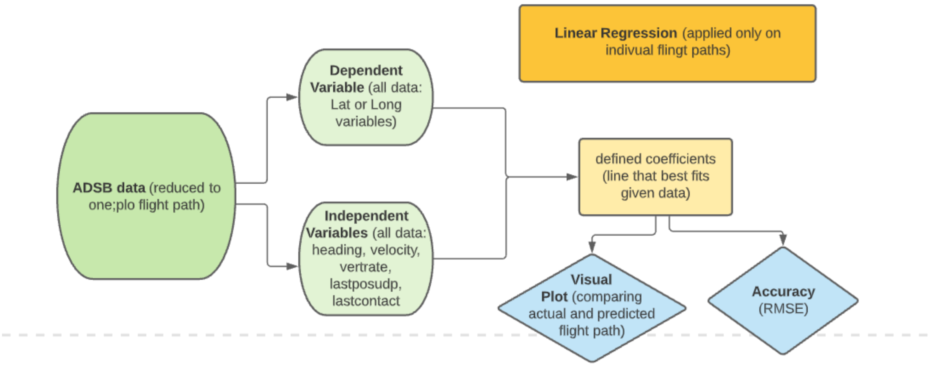

Linear Regression (applied only on indivual flingt paths)
Dependent Variable (all data: Lat or Long variables)
ADSB data (reduced to one;plo flight path)
Independent Variables (all data: heading, velocity, vertrate, lastposudp, lastcontact
defined coefficients (line that best fits given data)
Visual Plot (comparing actual and predicted flight path)
Accuracy (RMSE)

Figure 1. Data Flow Diagram

Based on research I have identified 2 possible Machine Learning (ML) algorithms that are suitable for predicting time series data (1) Long Short Term Memory Neural Network (LSTM) and (2) Linear Regression. The latter is a little easier to understand as it is uses predictors (independent variables) to predict dependent variables based upon the linear formula y = c +b(x). Since linear regressions only predict one dependent variable, I constructed two linear regressions (predicting latitude and longitude, one at a time) based on the following independent variables: velocity (m/s), heading (track angle measured up to 360 degrees), vertrate (vertical speed m/s), lastposupdate (the age of the current position), and the lastcontact (the last recorded time of the previous signal/point). Because of the nature of a linear regression (which constructs a line of best fit for the provided data), the linear regression was only used to predict one flight path at a time, and is not applicable to predicting other flight paths.

Meanwhile, LSTM neural networks are rooted in deep learning and "capable of learning and memorizing long terms dependencies" (Biswal, 2021). This strength in LSTM is very useful, since our data is time dependent (i.e. order matters since a plane's location is dependent on its past locations). Preliminary, results for LSTM models were trained on individual flight paths (see *Figure 2* for an example output), with the same independent variables tested from the previously mentioned linear regression. These variables were then used to predict latitude and longitude. In addition to training the LSTM based on individual flight paths, I have also succeeded in training an LSTM model based on 57 flight paths intersecting the Chicago airspace, and thus can make predictions for any flight observed within this vicinity.

Based on *Figure 1* you can see some of the preprocessing steps taken before training the LSTM. The flight data was separated into two groups: training data, any observations that occurred within the first 54 minutes of the dataset's time (timestamps occurring before 12:54:10 AM), and testing data, the remaining 6 minutes of observations. Once the model was trained, training and testing data were used to plot the predicted flight path alongside the actual flight path to visually compare the results. In addition, LSTM accuracy can be quantitatively assessed from the Root Mean Squared Error (RMSE) for entire the model and as well as for the testing data. The hope is to see is a similar RMSE value for the test data and the training data, and ideally at low values (all my RMSE values would be measured in decimal degrees since this is the unit of measure for my latitude and longitude). The smaller the RMSE the better the LSTM.

Results

Figure 2 depicts the predictions and the actual flight path for flight a198e5. This flight is arriving at Chicago Midway International Airport and thus most of its coordinates for the end of its flight are stationary. You can see this reflected in the graph on the right, where we see two dots. During this flight's last 4.5 mins it was sitting in a terminal unloading passengers, thus the blue point is really stacked consecutive points over time that do not move. The prediction (seen in orange) is within 1 decimal degree of the actual plane's location and is stationary but still reflects an acceptable accuracy error as seen in the RMSE values for both the latitude: 0.02564 decimal degrees and longitude: 0.03733 decimal degrees. For a general interpretation 2 decimal places is approximately equivalent to measurement in Kilometers (so 0.01 decimal degrees is about 1 km). Here, I can see my LSTM prediction for *Figure 2* is off by approximately 2.5 kilometers for latitude and 3.7 kilometers for longitude. I call these acceptable based on general flight regulations which stipulate planes should maintain 5 nautical miles (~9 kilometers) of horizontal separation when within controlled airspace and can minimize this to 3 nautical miles (~5.5 kilometers) when departing/arriving in airports (*Did You Know How close can a plane fly to another aircraft?*, 2019).

Thus, if my resulting predictions have a potential error ranging between 2-3 kilometers, we can still maintain a relative amount of safety (5 nautical miles).

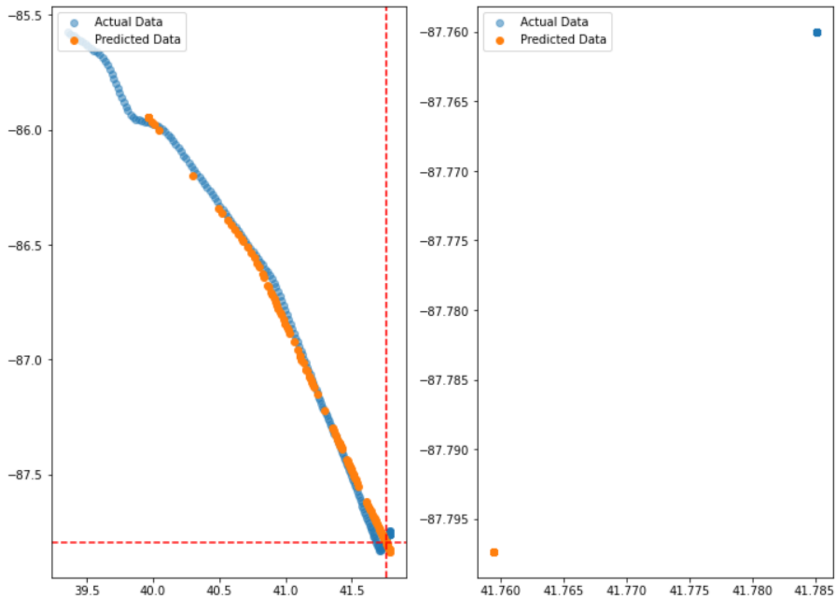


Figure 2: Flight path for a198e5. The left graph displays all the point data for a198e5, where blue is the actual flight path and orange the predicted flight path based on the LSTM. The red dashed lines are the markers that differentiate where the training data and testing data (the lower right quadrant was the testing data, while all the data elsewhere represents the training data). The image on the right is a close up look at the lower right quadrant which is based solely on testing data.

Looking at *Figure 3*, we can see the results from the linear regression predictions for the same flight used to train the LSTM in *Figure 2*. Based on the R^2 values for the linear regressions: $R^2 = 0.84$ for latitude and $R^2 = 0.91$ for longitude, I would surmise the overall fit for the data was good. Looking at the plot in the right-hand side of *Figure 3*, you can see the predicted flight path plotted against the actual flight path. Since linear regressions aren't divided into testing or training data, comparing the resulting prediction to those from *Figure 2* might be unfair (since a majority of the data in the plot for *Figure 2* to is based on training data), but still informative since they are predicting the same flight. A useful conclusion is that flight path a198e5 seems to follow linear path (as indicated by our R^2 values). Furthermore, RMSE values based on the predicted latitude from the linear regression was 0.30232 decimal degrees and longitude's error: 0.22191 decimal degrees. This is worse than the predicted values derived form the LSTM model from *Figure 2.*

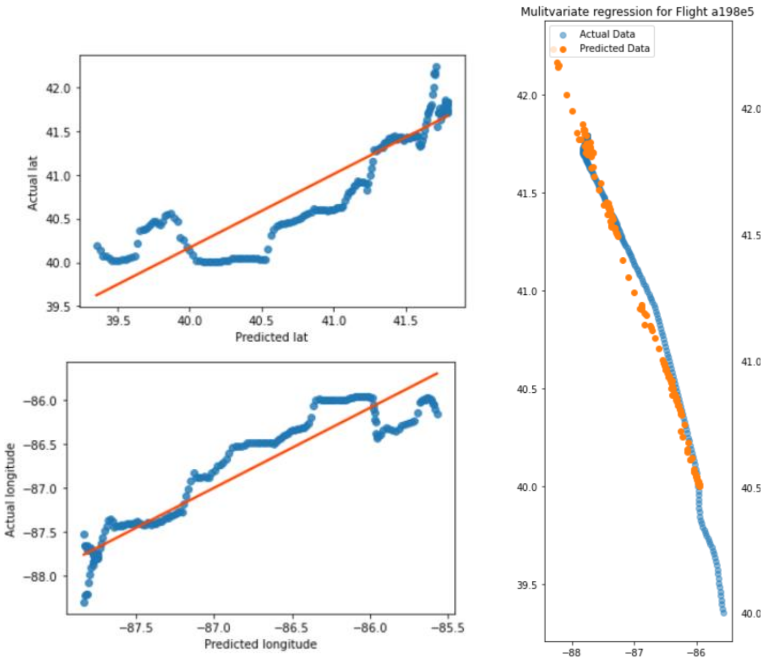


Figure 3: Flight path prediction for a198e5 (same flight path used in Figure 2). The plots on the right display a fitted line for the latitude and longitude (by comparing the actual data with those predicted by the linear regressions, we'd hope to see the data be as close to the red fitted line as possible). The plot on the right shows the results based on the linear regression predictions. The blue is the actual flight path, while the orange is the predicted flight path from the linear regression. As mentioned, this linear regression is specific to this flight and would be inappropriate to use to predict values on other flights (given that the lat and lon values would greatly differ.

Lastly, *Figure 4* displays the results from the LSTM trained on all 57 flight paths within the Chicago Airspace. The data had to be carefully reordered by flight path and then time (since order is preserved in LSTMs). As can be seen the blue points are the actual flight paths (keep in mind there are 57 different flight paths, discernable by the general linear nature of each flight), while the orange points are the resulting predicted values from the LSTM. I would like to take a moment to remind the reader that the LSTM was trained based on 90% of the actual flight data and is only new to ~10% of the data (which in the current depiction is indistinguishable from the training data). The advantage of this LSTM is that you can predict any flight within the trained lon and lat ranges. The resulting RMSE values were 1.60510 (approximately 170 km) for latitude and 1.029414 (approximately 112 km) for longitude. This is quite a significant jump when compared to the resulting RMSE from *Figure 2*, but I believe is explained by the fact that we have introduced a greater range for both lat and long (since we've introduce more flight paths, resulting in a greater spatial range of possible resulting predictions).

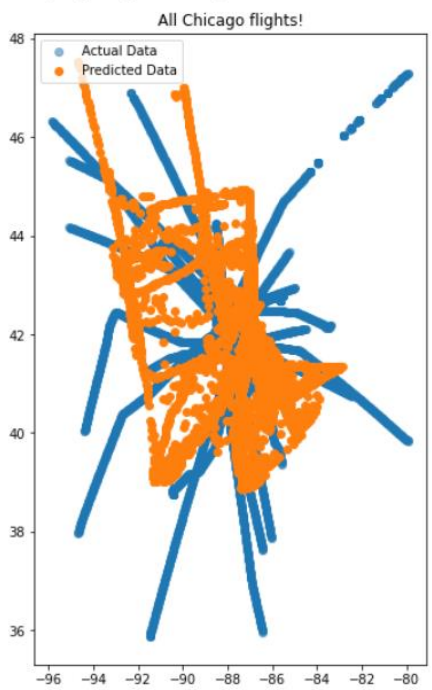


Figure 4: This plot displays all the point data for 57 flight paths flown around Chicago. The blue is the actual flight paths flown, while the orange the resulting prediction s made by the trained LSTM model (unable to distinguish each flight path, not able to differentiate the test from the training data-based predictions).

Table 1: RMSE errors.
Note that this table is a summary of the RMSE mentioned within the results, and all conversion from decimal degrees to kilometers are approximate and based on the general approximations that 0.1decimal degree = 11.1 km at the equator.

	Latitude RMSE (decimal degrees)	**Longitude RMSE (decimal degrees)**
LSTM (trained with flight a198e5)	0.02564 (approximately 2.5km)	0.03733 (approximately 3.7km)
Linear regression (trained with flight a198e5)	0.30232 (approximately 33km)	0.22191 (approximately 22km)
LSTM (trained with 57 flight paths)	1.60510 (approximately 170 km)	1.029414 (approximately 112 km)

Results Verification

Comparing the results for the linear regression and the LSTM based on flight a198e5 serve as a basis to identify initial performance. By comparing these two prediction outputs, I am able to discern if LSTM model training is any better than a simpler method (linear regression).

Discussion and Conclusion

Overall, I was excited to see results from the LSTM, but am disappointed with its lack of accuracy, especially that of LSTM from *Figure 4* (using 57 flight paths to train the LSTM model). More effort in isolating the resulting predicted flights from each other, is worthwhile, since at the moment, I cannot pinpoint what the LSTM predicted for each flight. I would like to see if it was able to capture the general linear characteristic that defines most of these flights, without having been expressly programmed as it is generally done so for linear regression models.

Now that I have successfully added more flight paths to train the LSTM, I would like to see how it would do if given more data (potentially continental spatial extant and over more time such as a week). I believe by adding more timed data observations, the LSTM will have more examples and a better ability to learn general flight lines. By introducing a larger spatial expanse, I expect it would increase prediction error, but think it is a practical step. Another idea is to reduce my predictions to a smaller time interval. In the above figures I predicted trajectories at most 6 minutes into the future, maybe I should focus on 1 minute into the future (about 6 timesteps) and see if the accuracy improves or at the very least is related to time (I want to know if accuracy decrease as time increases).

Given more time, I would then experiment with predicting the altitude as mentioned in the beginning of this semester project and integrating weather data. The intention is to track the plane in 3D space (since a plane's flight can interest in 2D space and not crash provided they are at different altitudes), and account for weather's effect on a flight's path, as well as current airspace. The challenge with adding weather data to the LSTM, is accounting for relevant weather data kilometers away from the plane's current position (since this can influence a flight path's course).

References

Biswal, A. (2021, October 28). *Top 10 Deep Learning Algorithms You Should Know in (2021)*. Simplilearn.com. https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm

Did You Know How close can a plane fly to another aircraft? (2019, November 11). BAA Training. https://www.baatraining.com/how-close-can-a-plane-fly-to-another-aircraft/

Sakyi-Gyinae, M. K. (2019). *A Machine Learning Approach to Evaluating Aircraft Deviations from Planned Routes* (p. 73) [Thesis]. https://repository.tudelft.nl/islandora/object/uuid%3A274b4386-539a-4193-80e9-f120c8d4832e

Self-score

Category	Description	Points Possible	Score
Structural Elements	All elements of a lab report are included **(2 points each)**: Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	**28**
Clarity of Content	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**.	24	**18**
Reproducibility	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	**22**
Verification	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**.	20	**20**
		100	**88**