Peregrine

Nathan Justice

DATS 6501 – Capstone

Final Project Report

Introduction and Background

Automatic Dependent Surveillance – Broadcast (ADS-B) is a burgeoning technology used to track and monitor airspace travel across the world. The goal of ADS-B is to bring global air traffic control (ATC) into the 21st century. Aircraft are equipped with a transponder, hardware devices called ADS-B Out, which broadcasts information about the plane through the air. The type of information sent out includes data describing the aircraft's identity, geographic location, altitude, and speed. The ADSB-B Out broadcast signal can be captured by receivers, hardware devices called ADS-B In. ADS-B In devices can be set up in on ground locations or on board other aircraft in order to facilitate airspace situational awareness and communication. Once deployed, ADS-B requires no human interaction to function. The broadcast information is freely sent unencrypted over the air, and anyone with a receiver is able to collect and use this information (Skybrary).

Operation of an individual ADS-B system has two main components, a certified global navigation satellite system for collecting data and an ADS-B Out unit for broadcasting data at 1090 MHz. Messages are sent as 112 bit long streams of data, which require decoders to parse

out relevant information (Junzi Sun). ADS-B is part of the United States federal program called Next Generation Air Transportation System (NextGen). NextGen is an ongoing effort to improve and modernize technologies that pertain to the safety, efficiency, and accessibility of transportation systems. The Federal Aviation Administration (FAA) set a mandate in 2010 declaring all aircraft must be equipped with an ADS-B Out device by January 1st, 2020 (e-CFR).

Problem Statement

The problem statement for this project is as follows: *Despite the abundance and accessibility, ADS-B data are often messy, incomplete, and difficult to use in their raw form. By implementing fixes to the data quality, useful tools can be developed to facilitate analysts' investigation into projects involving aircraft and airspace travel.*

Motivation

This project is done under the supervision of Data Machines Corp., a data analytic research and consulting company, who I am employed by as a research intern. The client for this project is the Center for Advanced Defense Studies (C4ADS). C4ADS is a group of highly technical investigative journalists and data analytics experts who use open data sources to report on global networks of nefarious actors. C4ADS has a partnership with ADSBexchange, an online community of flight enthusiasts. Together C4ADS and ADSBexchange distribute ADS-B In

receivers to interested parties throughout the world. People who purchase or are given receivers set them up and install ADS-B initializing and decoding software built by the ADSBexchange community. Users collect ADS-B information and transfer the data to ADSBexchange who assume the responsibility of warehousing and maintaining it. One of the projects that has gained C4ADS considerable notoriety is their effort at curbing wildlife trafficking by examining data in the air transport sector (C4ADS 2018). Working with ADSBexchange provided data has been fruitful for the C4ADS team thus far, however they are limited by quality issues in the data. Having access to more useful and useable ADS-B data would be a substantial boon to the efficacy of their work.

Problem Elaboration

The specific project goals addressed in the remainder of this report are three-fold:

1. Improve the quality of the existing ADS-B data and engineer new features woefully lacking in its current form
2. Develop an application, informed by user-stories, that will facilitate the way C4ADS analysts utilize ADS-B data
3. Implement features that allow client-specific questions to be investigated

Project Scope

This project began in June 2019, when I began working for Data Machines Corp. Client and user-story interviews were conducted in early July 2019. As a result of the C4ADS and ADSBexchange partnership, my working on this project gave me access to ADSBexchange's full backlog of ADS-B data. ADSBexchange started collecting data June 9th, 2016. Since then, ADSBexchange has processed petabytes of data each month. They curate and store around 10 gigabytes of ADS-B data each day and currently warehouse about 13 terabytes of data in an AWS Redshift database. The range of data collected approximates total global coverage more and more because as time goes on, more people in more parts of the world are setting up receivers to feed into the ADSBexchange pipeline.

Relevant Research

Due to the fact the majority of the direction, instruction, and background information for this project was provided by the client, C4ADS, very little research was needed to carry out the project goals. However, I did look for evidence to validate the quality of transmissions of ADS-B data, specifically to what degree the required amount of information is successfully sent and received. According to one report by Tesi and Pleninger in 2017, 86.42% of messages they analyzed satisfied federal quality requirements (Tesi & Pleninger 2017). I also focused on a publication by Syd Ali *et al.* that empirically investigated and documented recurring patterns of erroneous anomalies in the GPS data recorded by ADS-B In devices, particularly artifacts such as gaps and jumps in the recorded geographic coordinates (Syd Ali 2016). These patterns, while

not greatly affecting the results of this project, were factors I had to be continually mindful of while developing the methodology.

Dataset Description

Each record of ADS-B data in ADSBexchange's Redshift database represents a ping collected from an aircraft. A ping is a snapshot of 37 different features with descriptive information about the aircraft itself and where it was in space and time the moment the ping was recorded. As mentioned in the introduction, the ping data itself is actually a binary stream. In order to populate the Redshift table, the data is first processed using decoders built by members of the ADSBexchange community. Most of the 37 features are captured from the data stream directly, however some fields are derived values created during the encoding process or provided by individuals during the collection process.

In practice, the analysts at C4ADS aren't particularly interested in looking at a map with a sea of pings depicting all of the discrete locations a particular aircraft was observed. What they're truly interested in is looking at flights. They want to be able to analyze where a plane is coming from, where it is going, and changes in flight patterns over time. The ADS-B data as it is stored in the Redshift database has fields for whether or not an aircraft is on the ground as well as fields for source and destination. However, it is clearly articulated on the ADSBexchange Data Description web page that these fields are "based on user-reported routes, and [are] quite often wrong. Don't depend on [them]".

Whether or not an aircraft is on the ground and the source/destination of the aircraft at a particular time are the two most important pieces of information to the C4ADS analysts. These two attributes are fundamentally paramount to the ability to uniquely identify distinct flights that a given aircraft has made. The next three sections of this report describe my efforts to re-build this information in order to provide the C4ADS team data that is more useful to their work.

Data Collection

Working with C4ADS gave me unrestricted access to the ADSBexchange Redshift database. In order to repopulate the field for whether or not a plane is on the ground, it was necessary to have information about how high up an aircraft is in a given ping. ADS-B does record altitude. However, altitude alone is not enough information to determine how far off the ground an aircraft really is. Attitude is relative to mean sea level, therefore a plane landed at Denver International Airport would suggest it is still 5,430 feet in the air even when it truly is grounded. In order to control for mean sea level as the ground zero measurement, I had to first source a global digital elevation model (DEM). A DEM is a raster representation of elevation data. DEM's can be extremely large depending on their resolution. I elected to use 15 arc second tiles from Jonathan de Ferranti's Digital Elevation Data site. I then used Esri's ArcGIS software to resample (lower the resolution) and stitch together the tiles into a global raster layer with 450m resolution. The new DEM was just under 2 gigabytes large, small enough to fit in memory while still having relatively high accuracy. In order to determine the source and destination of a flight, I had to first procure a dataset of global airports. I used open source data

compiled by the online community at OurAirports. I had to do only minor alterations to the dataset before being able to put it into the development pipeline.

Data Preprocessing

The ADS-B Redshift data had a few issues with its structure that made it difficult to work with. In particular, for the values representing altitude, speed, trak, and vertical speed, the unit of measurement was determined by the corresponding value in another column. Knowing the end data product will be stored in a Postgres database, which uses a relational database management system, I decided to encode out the values by appending a new column for each variable and each corresponding unit of measurement. The new variable encoding system made it easier to query information from the Postgres tables. The next step in the preprocessing stages was to calculate the height field so it could be used when determining whether or not a plane was on the ground. This was achieved by a simple subtraction between the elevation from the DEM cell intersected by a given ping's GPS coordinates and the corresponding altitude. The OurAirports dataset represents airports as points, which couldn't be used when trying to determine the source and destination of a flight. To resolve this issue, I appended 10km buffers around each of the airports.

Data Exploration

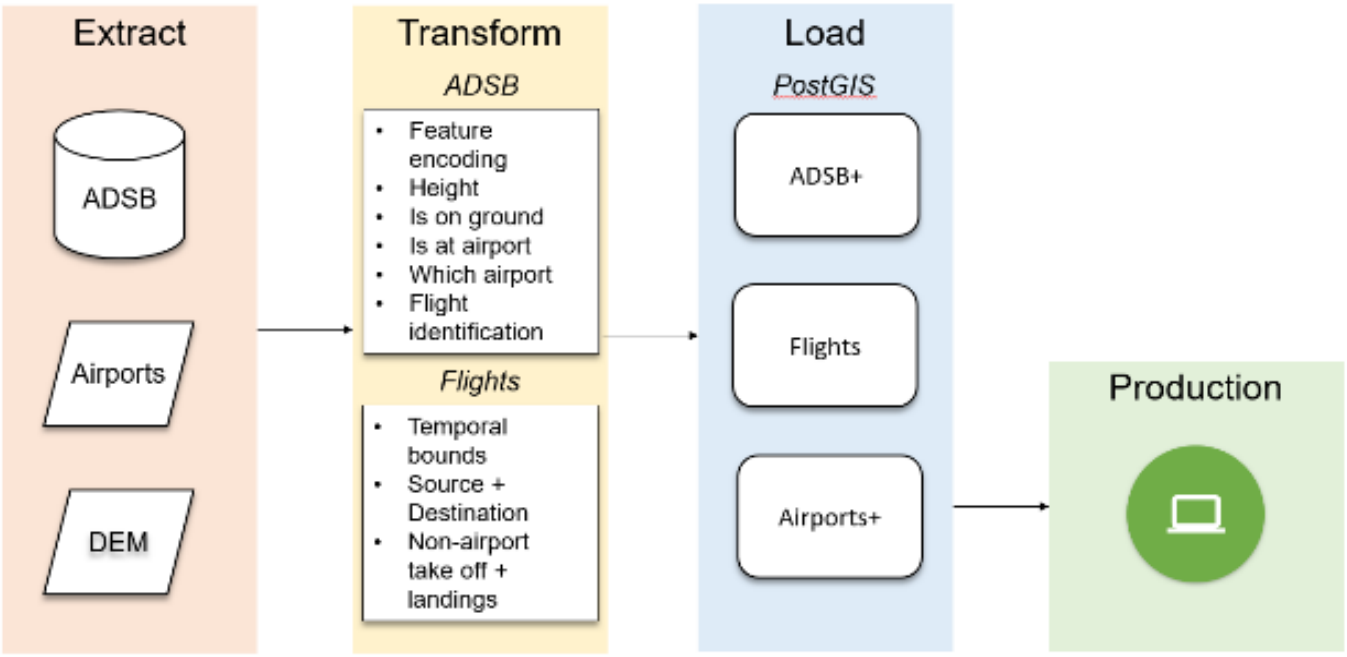
Using the augmented datasets, a colleague of mine worked some data science magic to solidify key threshold values for speed, height, and vertical speed. If all three of these threshold values were satisfied by a given aircraft observation, we could determine with a high degree of confidence that the aircraft was on the ground at the time of that observation. She also calculated threshold values for measured distance traveled and inferred distance traveled over time, which are were used during the unique flight identification process.

Feature Engineering

Armed with the new threshold values and the buffered airport dataset, I coded some simple algorithms to append information to each aircraft observation for whether or not it was on the ground, whether or not it was at an airport, if it was at an airport then which one, and a unique flight identification number. I then used the flight identification numbers to create a flights table where each record is a representation of the chronological first and last ping of the set sharing the same flight identification number.

Results & Analysis

The project as a whole can be described as one giant extract, transform, load (ETL) pipeline – which is summarized by the following image:



Data was pulled from the ADSBexchange Redshift database. Using the global airports dataset and DEM, each record was augmented and loaded into an enhanced ADS-B ping table. Additionally, and most importantly, a new flights table was created as well. The creation of the flights table was the most important aspect of this project because this data, ADS-B represented as flights and not pings, did not exist previously. I selected Postgres as my load storage system because Postgres databases can be equipped with the PostGIS plugin, a powerful extension that allows complex geospatial queries to be performed on relational data. The application that I developed leverages the geospatial querying capabilities extensively.

Conclusion

Using the mountain of ADS-B data at my disposal, I developed and implemented a data pipeline that accomplishes two valuable tasks: 1) the pipeline cleans messy ADS-B data and organizes it into a more usable form, and 2) the pipeline utilizes novel heuristics to translate

noisy ping-level data into more useable and informative flight-level data. It's important to note that each record in the ping-level data can be mapped to a record in the flights table, and a given flight observation can be decomposed back to the set of original pings that comprised it. This is an important aspect of the data because it means the analysts at C4ADS will have an easier time managing the provenance of their data throughout their various investigations.

A lot of effort by me was put into designing and building this data pipeline. However, the vast majority of my time spent working on this project was dedicated to the map-based web application I built for the C4ADS analysts to interface with the newly curated data. The app was developed using Meteor.js, MongoDB, and Mapbox.js on the back-end, and React.js and Material design on the front-end. The application was also equipped with an encrypted account authentication system and other fundamental security measures. The purpose of the application was to allow users to query the new data by graphically building queries based on the relevant information of a given investigation and organize the results in a structured and reproducible way. After the data is specified and retrieved, users can then explore the results using either a map or tabular interactive view. The app is currently deployed into production at this web address: http://ec2-35-172-177-205.compute-1.amazonaws.com

Project Limitation

The limiting factor of this project was computational power. Transforming even a small subset of the data took a substantially large amount of computational power, largely a result of the geospatial data and operations used. I spent about a month explored using Amazon Web Services (AWS) Glue as a high performance computing environment for the pipeline. Ultimately, I was halted by seemingly unresolvable obstacles. The first of which is a product of the way

AWS manages Virtual Private Clouds (VPC). The VPC's confine communication access such that I was unable to query the ADSBexchange Redshift database directly because it was external to the environment. The only way I could think to resolve this issue would be to make a copy of the Redshift database under the C4ADS environment, which would be a huge and shockingly wasteful cost to bear. Additionally, AWS Glue uses a PySpark environment that is not conducive to installing new packages, which was an issue because I was unable to use the geospatial Python packages necessary to determine airport source and destination. Lastly, AWS Glue is restricted to tabular data structures, making it impossible to natively use the DEM, which was necessary for determining whether or night an aircraft is on the ground.

Future Research

Despite the challenges, the pipeline does need to be hosted in an automated, high performance environment and run on ADSBexchange's historical backlog before this project can be considered fully completed. Another necessary future development is to implement a system that can handle automated batch jobs because new ADS-B data is added to the Redshift database every night. Additionally, it would be advantageous to have a system that continually refines the heuristics used to determine on ground and unique flight identification as more and new data is passed through the pipeline. Lastly, and as always will be the case, there are endless new features that can be added to the app to make it a more robust, useful, and productive tool to use.

References

C4ADS. 2019. In Plane Sight: Wildlife Trafficking in the Air Transport Sector. https://static1.squarespace.com/static/566ef8b4d8af107232d5358a/t/5b8847a8c2241b8686fcd05a/1535657953438/In+Plane+Sight.pdf.

e-CFR (Electronic Code of Federal Regulations). §91.227 Automatic Dependent Surveillance-Broadcast (ADS-B) Out equipment performance requirements. https://www.ecfr.gov/cgi-bin/text-idx?node=14:2.0.1.3.10#se14.2.91_1227.

Junzi Sun - https://mode-s.org.

Skybrary Wiki
https://www.skybrary.aero/index.php/Automatic_Dependent_Surveillance_Broadcast_(ADS-B).

Syd Ali, B., Schuster, W., Ochieng, W. et al. (2016). Analysis of anomalies in ADS-B and its GPS data. GPS Solut 20: 429.

Tesi, S., & Pleninger, S. (2017). Analysis of Quality Indicators in ADS-B Messages. MAD-Magazine of Aviation Development, 5(3), 6-12.