Analysis and visualization of comparison between air cargo and passenger flights

Yiming Xu
2696855
Vrije Universiteit Amsterdam
yiming.xu@student.vu.nl

Yongqing Liang
2721589
Vrije Universiteit Amsterdam
y4.liang@student.vu.nl

Haohui Zhang
2722930
Vrije Universiteit Amsterdam
h17.zhang@student.vu.nl

ABSTRACT

More and more aircraft use ADS-B to broadcast position and status information. This process produces a large amount of data to help researchers study the aircraft's route planning and flight characteristics. However, the cargo and passenger attributes of commercial aircraft are not marked in the ADS-B code. Therefore, it is difficult to directly use the ADS-B code database to analyze and compare the flight characteristics of passenger and cargo aircraft. In this project, we referenced external data sets and machine learning classification models to identify the cargo or passenger types of all commercial aircraft, clustered the main routes of freight and passenger transport, and then compared the differences between the two aircraft routes. We also counted the passenger and cargo transportation at European airports and analyzed the market changes of cargo aircraft in the past two years.

Keywords : ADS-B, Air Cargo, Classification, Route Cluster, Big Data.

1. INTRODUCTION

Automatic dependent surveillance-broadcast (ADS-B) technology [1] is an aircraft surveillance technology that records and encodes information such as latitude, longitude, speed, and flight altitude of an aircraft at a certain moment. At present, more and more aircraft adopt this technology and generate a lot of information. These messages are not encripted and can be received and decoded by anyone. Enthusiasts around the world collect these information to make it available to the the public which is enormous.

OpenSky Network is an organization that collects, processes and stores ADS-B data generated by air traffic. We obtained one week's ADS-B message for 2016 and 2017. The original data set is about 800GB.

In this project, we focus on identifying cargo aircraft and passenger aircraft from all data. In the ADS-B decoded data set, these data are mixed together, and there is no attribute that can directly distinguish them. After identifying the two types of aircraft by integrating with outsources data and being classified by random forest model, we analyze and compare the differences in flight routes and frequency of the two types of aircraft. After that, we used the clustering method to find the main routes of passenger aircraft and cargo aircraft. Finally, by comparing two years of data and integrating external data, we observe the changes in the cargo aviation market, and summarize the route of cargo flights and the characteristics of cargo aircraft.

2. RELATED WORK

2.1 OpenSky Network

OpenSky is an open source network that brings together ADS-B information collected by volunteers from around the world for the purpose of air traffic statistics [2].

There are some limitations to the data collected by the OpenSky.

- The quality of the data is influenced by the density of ground sensors. In areas where coverage is poor or non-existent, it is not possible to detect complete aircraft information.
- Information is collected by volunteers and there is no guarantee of the reliability of the information collected.

2.2 Random Forest Model

Random forest is a classification algorithm composed of many decision trees. It uses bagging and feature randomness when constructing each tree to try to create a forest of unrelated trees, and its committee's prediction is more accurate than any tree's prediction.[3] The random decision forest corrects the habit of the decision tree overfitting the training set. However, data characteristics can affect their performance.

The first algorithm of random decision forest was created by Tin Kam Ho[4] in 1995 using the random subspace method. The extension of this algorithm was developed by Leo Breiman[3] A furthermore extension combined Breiman's "bagging" idea and random selection of features was first introduced by Ho[4] and later independently by Amit and Geman[5] to build a set of decision trees with controlled variance.

2.3 Route Distance

In order to identify the main routes, the most important thing is how to go about measuring the similarity between the two routes.

Considering that each route has more or less track points, we are not able to apply the traditional distance algorithm. First, we approximate the latitude and longitude coordinates to one decimal place, then remove the duplicate position information and convert the latitude and longitude information of each track point into a string, which is equivalent to assigning a number to each sector of the map of approximately 15km*15km. This means that the representation of each route is converted into an array of strings, and the similarity between routes can then be found using the minimum edit distance[6] .

3. RESEARCH QUESTIONS

3.1 Questions

The main objective of this project was to create a visual web page that could be interacted with and statistically study some of the characteristics of cargo flights.This raises some questions and problems regarding the project.

- How can information about flights be extracted from the raw data set?
- How can each flight be identified from the many discrete trajectory points?
- How to identify the departure and landing airports for each flight?
- How to measure the similarity between aircraft routes so that the main routes can be identified?

For visualisation, we need to use static web pages on a computer with relatively low computing performance for presentation

- How can the size of the dataset be reduced so as to visualise it more smoothly without using too many computational resources?

We use the class of carrier resolved from Callsign as the class of aircraft (passenger or cargo), but some carriers do both cargo and passenger traffic.

- how can we distinguish between them?

3.2 Technologies

In order to find the answers to the above questions in the data, we need to choose the right algorithm. We need to consider two aspects: the reliability of the algorithm on the one hand and the performance of the algorithm on large data sets on the other.

- The reliability of the algorithm: We decided to download some small datasets locally and then write algorithm code that would test the reliability of the algorithm and allow for fast iterative improvement of the algorithm. To do this, we used the Python data analysis technology stack, namely Numpy, Pandas, Sklearn, Matplotlib, etc.
- Computational performance: We use the Scala Spark and Java-adsb packages to decode and analyse data on large datasets, because of parallelism, they are faster and more efficient.

Our work is divided into four parts, data preprocessing, flight classification, route clustering, analysis and visualization. The pipeline of the project is shown in Figure 1

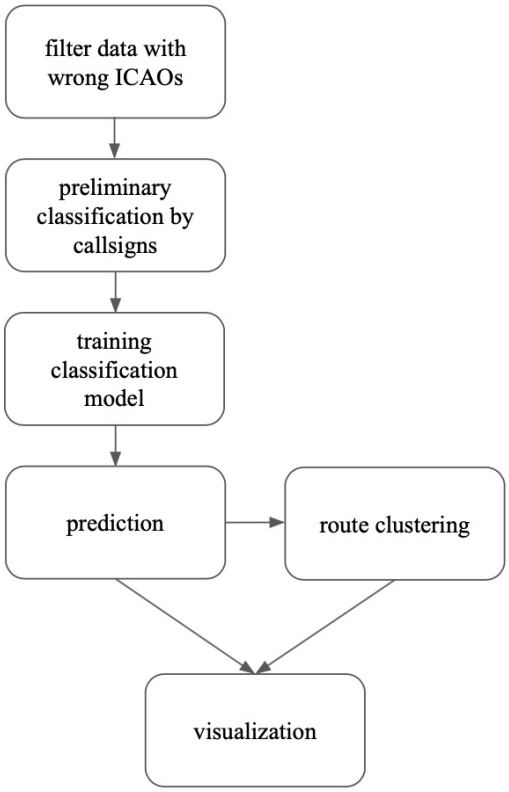

Figure 1: The prototype of visualization.

4. PROJECT SETUP

4.1 Project Outline

This project have incremental development by using Agile project management method 'Scrum'. There are three phases including initial phase (Outline planning), a series of sprint cycles (Assess, Select, Develop and Review) and the project closure phase. Because of the limited time, we separate our project into two iteration.

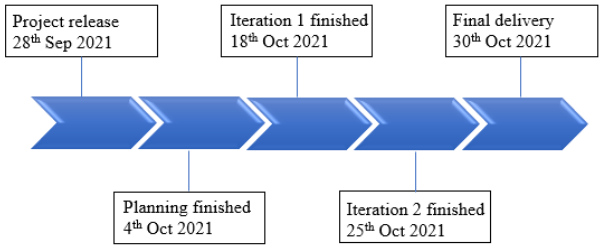

Figure 2: Time Schedule

The initial phase mainly focusing on the entire project outline planning and requirements finding. In the first iteration, we will focus on the data cleaning, track detection, preliminary data classification of aircraft types, data transfer and data dynamic visualization according to the classification. In the second iteration, we will put our concentration on classifying the aircraft types as well as finding feature through machine learning model, clustering the main route,

Protocol	Count
Short identify reply	59000
null	8
Long air-air ACAS	12754
Short air-air ACAS	135256
Short altitude reply	159046
Mode-S Extended Squitter (ADS-B)	235694
Comm-B altitude reply	166659
All-call reply	526316
Comm-B identify reply	77189
Military Extended Squitter	3831

Table 1: Protocols of rawMessages and Count

Type	Count
Identification messages	12675
Airborne velocity messages	103720
Operational status reports (airborne and surface)	3162
Airborne position messages (including global and local CPR)	104950
Target state and status messages	6021
Aircraft status reports (emergency/priority, TCAS RA)	1118
Surface position messages (including global and local CPR)	523

Table 2: Type of ADS-B information and Count

data comparison and data exploration to seek the value behind large-scale data. In the project closure phase, we will complete the documentation required for the project and summarize the project experience.

4.2 Understanding The Data

Insight into the data is necessary before data processing can take place. So we decoded an avro file and counted the number of various types of Mode-S message as shown in Table 1.

Of the above message types, the ADS-B Message is the most informative for our project.

The ADS-B standard specifies different types of messages to be sent at different frequencies, among the various subtypes of ADS-B Message, there are two types of information that are very useful for our analytical work:

- Aircraft identification messages contain the callsign of the aircraft, which can be used partially separate cargo flight from passenger flight
- Airborne position messages contain altitude, latitude and longitude of the aircraft

Each ADS-B transmitter is assigned a unique 24-digit ICAO number. As the transmitters installed on the aircraft hardly change, we can safely assume that an aircraft corresponds to a unique ICAO number, which is also recorded in each raw ADS-B message. Therefore, we can associate ADS-B messages to individual aircraft.

From Table 2, we can calculate that if we extract only the information in the original dataset that is useful for our project, we can reduce the amount of data by approximately 91.5 %

4.3 Parsing Extraction Data

First of all, we set up a process to read the data and parse relevant information. The data available to us consists of the raw transmitted ADS-B messages, stored in avro and csv format. Each data point contains a rawMessage as well as the timestamp when receiving antenna. The rawMessage is the mode message field in hex representation which is relatively complete original information [7]. We parse the rawMessage by using java-adsb8 library provided by OpenSky to get the ICAO 24-bit address, position contains latitude, longitude and altitude ,as well as the callsign we mentioned in the previous section. In order to use java-adsb in Spark, we need to modify the code to serialize Scala objects into text or binary data to support persistence or network transmission. Besides, we need to construct user defined functions to process the dataframe (abstraction on Spark's Resilient Distributed Data Set) later, the code still need to be serialized so that it can be sent to each worker node at execute on its data segment. In order to reduce the size of the transmitted data, no rawMessage contains all the information about the current aircraft position. The messages are separated in odd and even position frames, it need both two frames to pinpoint the exact position given in the raw message, which means we need two messages to get the position: one is in odd position frame, the other is in even position frames and the frame difference is less than 10 seconds. To parse the data, we do the following:

- Extract the ICAO, callsign and position format from the raw message, using java-adsb.
- Group the data by ICAO and timestamp.
- Pass the raw messages of one ICAO in chronological order to resolve the position information.

This extraction method has considerable scalability, because we order the datas by ICAO and each aircraft can be extracted in parallel. Using this parsing process, we can flexibly extract the data of interest. We use this scheme for different types of extraction. For instance, extract the data of a specific airplane, etc.

There is a situation that the data contains outliers, which will crash the execution and drop exception. In order to make our script robust to failures due to corrupted data, we add try-catch method to catch exceptions in all process to ignore those records.

4.4 Data Preprocessing

The purpose of the data preprocessing stage is to reduce and reorganize the data set into a smaller format that can be used for classification and visualization. A pipeline of data processing elements is produced in this section. Each element is a Spark application that usually runs on a cluster, unless the data set is small enough to run in the local environment.

4.4.1 Track Detection Algorithm

An aircraft may take off and land several times and its trajectory points are discrete, so a track detection algorithm is needed to segment the trajectory points for each flight.

We need to write some rules as markers to distinguish between two adjacent flights:

- A time difference of more than 20 minutes between two adjacent track points

- Or an altitude difference of more than 1,500 feet between two adjacent track points
- Or a distance of more than 40km between two adjacent track points
- Or when the altitude is below 3,000 feet and the aircraft's altitude tends to fall before it rises

Each set of trajectory points is further tested for trajectory plausibility using a number of rules:

- The first trajectory point is less than 3000 feet
- And the last trajectory point is greater than 3000 feet
- And shows an upward trend at the beginning of the trajectory and a downward trend at the end of the trajectory

The end result is shown in the diagram 3.

We acknowledge that the above filtering rules are strict and may throw out some usable data, but given the unreliability of the raw data and the complexity and variability of real aircraft trajectory points, we would like to be able to use higher quality data for the rest of the analysis.

For the 2016 and 2017 datasets, we found a total of 13,790 routes. Surprisingly, this didn't take too long, about 2 hours, probably because a lot of data of a type that was not informative for our project was filtered out at the beginning.

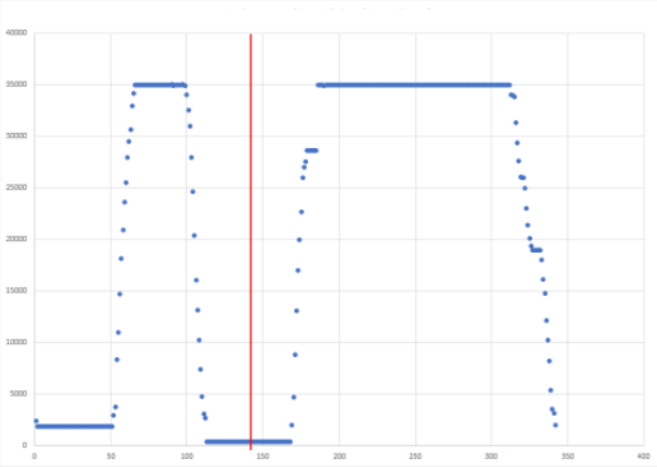

Figure 3: Two flight splits

4.4.2 Airports Detection

In order to determine the departure and landing airports for each flight, we found an external dataset containing information on the coordinates of many airports from OurAirports[8]. We used the very simple approach of identifying the airport closest to the first trajectory point of the route as the departure airport for that route and the airport closest to the last trajectory point as the landing airport.

Of course, when using external datasets, we filtered out closed airports and helipads, leaving commercial airports for use in our airport detection algorithm.

We are aware that this can lead to problems, for example some airports are close to each other and can misjudge the take-off or landing airports for the route. But given the data available to us, and the discrete and somewhat unreliable raw data, this is the best we can do at the moment.

In the results, we found that some routes had the same departure and landing airports, which could be due to We did not succeed in splitting the departure and return flights, Probably because their track points don't have an obvious signal for our track detection algorithm to work. In the statistics we have eliminated this situation and improved the reasonableness.

4.4.3 Linear Interpolation

Because of the unreliability of the data quality, not every timestamp in the data we get has position information, and to facilitate the visualisation, we have to interpolate the aircraft's trajectory point information. Considering that the aircraft broadcasts its position information every short time interval, we use linear interpolation to get a good reduction of the original flight trajectory of the aircraft.

Since Spark does not have the function to interpolate values in order of timestamp, we need to implement this manually

1. calculate the maximum and minimum values of the interval formed by adjacent timestamps
2. find the slope and use it to fill in the latitude and longitude values in a progressive or decreasing manner

Of course, we also approximate the timestamps, for example by using 1000s as the base time unit, so that the flights can be interpolated with less data and thus easier to visualise. As the visualisation is done on the whole European map, the effect is not seriously affected.

4.4.4 Data Filtering and Cleaning

The transport types of flights (passenger aircraft or cargo aircraft) are not clearly identified in the dataset, so they are not easy to distinguish. Accurately identifying the transport types of these flights is the first challenge we have to face. We decided to use the first three digits of the callsign code decoded by raw messages to find out which airline the aircraft belongs to. We divide the aircraft sources in the dataset into four categories, military aircraft and private aircraft, cargo airline aircraft, passenger airline aircraft, and passenger-cargo mixed airline aircraft. Since some airlines such as European Air Transport and Air China Cargo only focus on cargo business, aircraft from these airlines can be easily classified as cargo flights, we find this type of data from AirlineCodes [9]. Similarly, we find those aircraft from airlines that only focus on passenger transportation from Wikipedia [10]. For those aircraft from airlines that are engaged in both passenger and cargo business, we cannot directly judge their category, and will use supervised learning to predict their category later.

Because the transmission problem and try-catch method, the data parsing from raw messages may contain null and outliers. All the features we have so far are critical and even decisive for our subsequent processing, classification, visualization, and statistical analysis. Therefore, all missing data will be filtered out except that the callsign and height are empty tolerable.

Before training the model, it is necessary to detect and process outliers. For the distance and time feature, in order not to affect the data distribution, we replaced the measured

values whose deviation from the mean was more than two standard deviations with the default values. High outliers that deviate more than three standard deviations from the mean are discarded [11]. As for the max latitude feature, we replace the outliers with the highest altitude a plane can reach, which is 6000 meters.

4.5 Aircraft Type Classification

After the preliminary classification, nearly half of the dataset is successfully classified. The reason is that our passenger airline data does not cover all the airline in the world, as well as there are a large amount of airline company provides both the cargo and airline service such as Deutsche Lufthansa. In order to get more accurate result, we decided to use machine learning model to classify the two type of aircraft. The feature importance can also be found through the training process. In this process, we only need the data point related to the departure and landing of each flight.

4.5.1 Feature Increment

A feature is an important characteristic presented in data, usually obtained through the calculation, combination or transformation of attributes. Good quality data and features are often the foundation of a well-performing model. After the data preprocessing process, three important feature can be used to classification, separately the absolute flying distance, the flying time, the max altitude of a flight. These three feature is not enough to classify the flight. Thus, we need to increase the feature.

First, we get average speed, a derived feature which obtained by dividing distance and time. After analysis, another valuable feature is whether it is day or night when the plane is flying. To get this feature, we use the Astral package to calculate the sunrise and sunset time. All the time message we decoded are in the UTC time zone, so it only need the position(altitude and longitude) and the date to calculate the sunrise and sunset time. The process is as follow:

- Extract the timestamp, altitude and longitude of each coordinate point.
- Parse the timestamp into date.
- Calculate the sunrise and sunset time.
- Compare them with the flight departure and landing time. If the time in the range of daytime, set 1, otherwise set 0.
- Find difference between departure and landing time and the sunrise and sunset time.

Now we get another six features which are whether departure time is at day or night, whether landing time is at day or night, the difference of departure time with sunrise as well as sunset time and the difference of landing time with sunrise as well as sunset time. Since we need to compare the feature importance to conclude the difference between cargo and passenger flights later, we do not choose to add dimension by basic conversion to a single variable.

To facilitate the comparison and weighting of indicators of different units or magnitudes, we normalize the data into a common scale. We use min-max normalizer to linearly rescale each feature to the [0,1] interval, which is done by shifting the values of each feature so that the minimal value is 0, and then dividing by the new maximal value.

4.5.2 Indicators

Another important preparatory work is to find one or multiple appropriate indicators. After analyzing the data, we find out that the ratio of passenger flights and cargo flights is more than 10:1, which is a very unbalanced dataset. In this case, we cannot use the basic indicator to determine the model performance. Although the precision and recall are very important indicator, just looking at one or the other cannot provide a complete prospect. We can achieve excellent accuracy when the recall rate is poor, or we can have poor accuracy when the recall rate is excellent. One of the method which can combine two measures is f-score. F-score is the harmonic mean of precision and recall[12]. In our case, because of the same importance of precision and recall, we set β=1.

$$F - score = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{\beta \cdot Precision + Recall}$$

Receiver operating characteristics curve(ROC) is a curve that measures the performance of binary classification system with false positive rate as the horizontal axis and true positive rate as the vertical axis. There is often a category imbalance in actual data sets, where there are many more negative samples than positive ones (or vice versa), and the distribution of positive and negative samples in test data can change over time.[13] The reason why we choose ROC curve as a indicator is that in the unbalanced dataset situation, the curve can remain constant and consider all the possible thresholds for the classifier.

4.5.3 Data Separation

Before training the machine learning model, we need to divide the whole data into training set and testing set. We use the train_test_split function from the sklearn package to accomplish that. However, our dataset is an unbalanced dataset, and in common situation if the imbalance ratio is greater than 4:1, the classifier is biased toward large categories. We solve this problem from the perspective of data sampling. We choose to compress the data of major categories using undersampling of data, which balances uneven data sets by keeping all data in the minority class and reducing the size of the majority class. Different sampling ratios should be adopted for different categories, but generally not 1:1, because it is far from the reality. In order to avoid increasing the bias of the model as much as possible, the specific sampling ratio needs to be determined through experiments, which will be elaborated in the next section.

4.5.4 Machine Learning Model

After getting the training set and testing set, we need to find the best possible model to train our data. We try several typical machine learning model such as support vector machine, decision tree, random forest tree, gradient boosted decision tree and multilayer perceptron and do a lot of experimentation. By comparsions of experimental results in Table 3, we find that the Random Forest model has a significant improvement in performance no matter comparing f-score and accuracy.

Random forest consists of a large number of individual decision trees that operate as an ensemble. The important problem of decision trees is that they are easy to overfit training data. The idea behind random forests is that each

ml model	10:1	5:1	4:1	3.5:1	3:1	2.5:1	2:1	1.6:1	1.3:1	1.1:1	1:1
SVM(linear kernel)	0.0000	0.0211	0.0157	0.2592	0.0309	0.3114	0.2834	0.3876	0.4626	0.3840	0.4489
SVM(kbf kernl)	0.0130	0.0497	0.0628	0.0419	0.0741	0.0958	0.1842	0.0914	0.1428	0.2026	0.3160
Decision Tree	0.4587	0.5495	0.5325	0.5352	0.6102	0.5323	0.5535	0.5152	0.5268	0.5105	0.4858
Random Forest	0.6822	0.6695	0.7027	0.7456	0.7197	0.6645	0.6646	0.5917	0.5707	0.5800	0.5362
Gradient Boosting	0.2174	0.4088	0.5172	0.4378	0.5405	0.5201	0.4965	0.6177	0.5515	0.5747	0.5514
Multilayer Perceptron	0.1844	0.3619	0.3809	0.3526	0.4209	0.3882	0.3823	0.3975	0.3266	0.3138	0.3437

Table 3: F1-score of multiple model

ml model	10:1	5:1	4:1	3.5:1	3:1	2.5:1	2:1	1.6:1	1:1
SVM(linear kernel)	0.8836	0.8232	0.8288	0.7969	0.7863	0.7273	0.6539	0.6535	0.6193
SVM(kbf kernl)	0.8851	0.8605	0.8545	0.8612	0.8423	0.8349	0.8587	0.8127	0.6397
Decision Tree	0.9072	0.8742	0.8457	0.8328	0.8549	0.8098	0.8052	0.7739	0.7292
Random Forest	0.9505	0.9179	0.9189	0.9119	0.8791	0.8863	0.8724	0.8574	0.5476
Gradient Boosting	0.8904	0.9024	0.8906	0.8672	0.8748	0.8568	0.8337	0.8527	0.7453
Multilayer Perceptron	0.8935	0.8778	0.8388	0.8368	0.8107	0.8185	0.7824	0.7856	0.7043

Table 4: Accuracy of multiple model

tree is likely to predict relatively well, but some data may be overfitted. If we construct many trees, and each tree has a good prediction but is overfitted in a different way, we can average the results of these trees to reduce the overfit. This model is also effective dealing with the unbalanced data [14].

After finding the model, it also needs to determine the distribution of the data, which is the ratio of cargo and passenger flights in training dataset. We locate the most befitting 5 ratio through the comparsion of f-score, and then construct the ROC curve to determine the ratio. Those five ratios have the same threshold. Area under the Curve(AUC) represents the degree or measure of separability. The closer a result from a contingency table is to the upper left corner, the better model predicts. The higher AUC, the better model performance. From the Figure 4 we figure out the best ratio is 3.5:1. The confusion matrix of 3.5:1 ratio is presented in Figure 5.

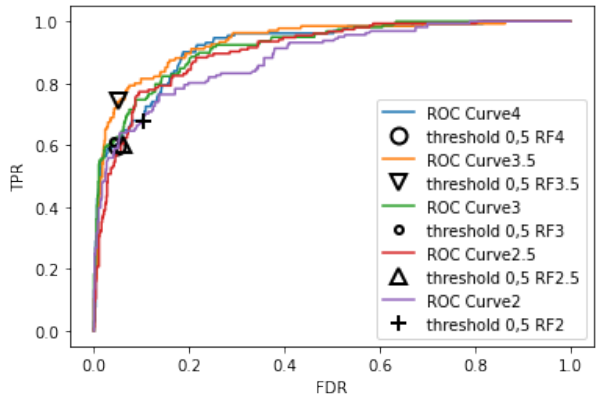

Figure 4: ROC curve of random forest model

4.5.5 *Prediction*

The important parameters that need to be adjusted in our model are n_estimators and max_features. To reduce overfitting and get a more robust integration, we need to average as many trees as possible, so we set n_estimators to 10000. Max_feature determines the randomness of each

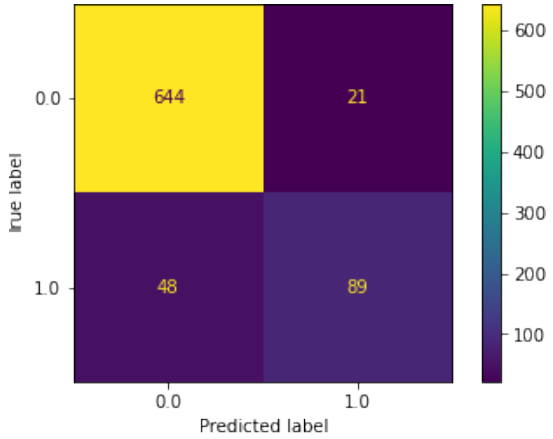

Figure 5: Confusion matrix of ratio 3.5:1

Year	Passenger flight	Cargo flight
2016	4566	358
2017	8246	618

Table 5: Final classification result

tree. Through experiments, we set this hyperparameter as the default value. The accuracy of the ultimate model is 0.9189. The f-score is 0.7456. The prediction we get is in Table 4.5.5.

The reason why the best accuracy and f-score are so unsatisfactory is that the model and data still have a lot of defects. First, we only have 14 days flight data, and the cargo data accounts for less than 10 percentage. Because of the limited amount of cargo data, the cargo data in test dataset is very small. It is difficult to induce effective classification results with limited data. One of the most effective solution is to expand our data. In addition, the best imbalanced ration we found is still very high. This is likely to lead to the biases towards large categories and cause the overfitting of the model. What's more, the valid features we found are insufficient, it seriously affect the performance of our model. It can explicitly observed from the confusion

matrix that there are many of cargo flight misclassified as passenger flight. It can conclude that the features we get are not informative enough to distinguish two types. However, if considered from another point of view, it can also indicate that there are quite a number of passenger and cargo flight are not significant different from each other. From the model perspective, more effective features are needed to train a model with better performance.

4.6 Main Route Clustering

In order to compare the route differences between passenger and cargo aircraft, we need to find the main routes by clustering all routes with the same take-off and landing airports. Surprisingly, there are generally multiple main routes, in which we find how many of the multiple main routes are for cargo aircraft, and obtain relevant conclusions.

4.6.1 *Conversion Route Representation*

In order to find the main route between a pair of airports, we need a way to measure the similarity between routes and thus find the routes which are most similar to all the other routes. The number of track points on a route needs to be reduced. One flight has many track points that are not easy to handle. So first we approximate the longitude latitude height to one decimal place, this means that a latitude and longitude coordinate can represent roughly a 15km*15km square. Then remove the duplicate track points, then build the route representation in the form of string arrays, each item of this array is a summation in the form of a string of latitude and longitude, as a unique representation of the square.

4.6.2 *Minimum Edit Distance*

Then, we calculate the distance between routes with the minimum edit distance.

This distance is represented by the smallest operands, i.e. insert, delete and put - transforming route 1 into route 2 and vice versa. Different weights can also be applied to different operations.

Normally the minimum unit of calculation for Edit Distance is a character, but we have changed the situation here and the minimum unit of calculation is an element of the array, which is a string representation of latitude and longitude.

It is not suitable to use the general distance algorithm, because the number of trajectory points of each flight is not the same.

4.6.3 *Clustering method*

For all flights at a pair of airports, we define the main route as the route that is the closest to the sum of all other routes.

When comparing the main routes for passenger and cargo traffic, we did not choose to find the main routes for passenger and cargo traffic separately. This is because the number of cargo flights is relatively small, with only one or two at most airports, so the results after clustering would be more biased.

Another advantage of using this algorithm is that it can be applied on large data sets, as we can collect the strings for each flight, put them in an array, put them in a row, and then use groupByKey and UDF to calculate the distance between routes, which can be parallelised and easily applied on large data sets.

4.7 Data Visualization

Our visualization mainly includes three modules: real-time passenger and cargo aircraft distribution map, passenger and cargo airports connection graph, and passenger and cargo volume comparison diagram within two years. These modules are mainly implemented through D3.js [15]. It requests vector geographic information in the form of GeoJSON, and uses Mercator projection to present them in SVG in first and third module. And in second module, D3.js provides hierarchical edge bundling methods to help us show the connections between different types of airports. The project visualization is shown in https://lsde-group16.github.io/Air-Cargo/ .

4.7.1 *Real-time Aircraft Distribution Map*

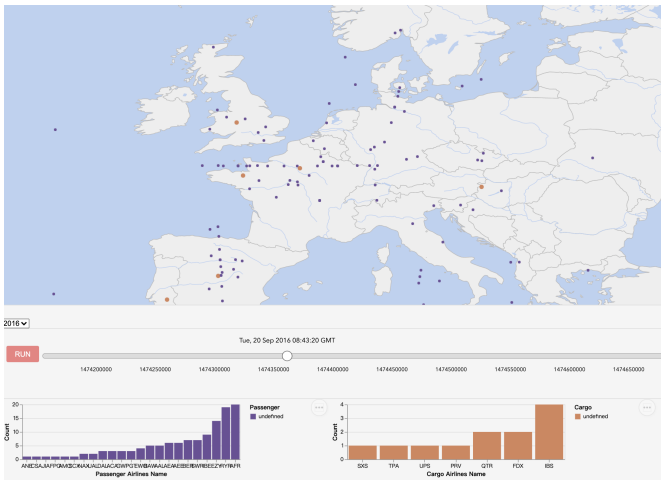

Figure 6: Real-time Aircraft Distribution

In Figure 6, We show the distribution of passenger and cargo aircraft on the map at each moment in one year at a speed of 1,000 times. In the previous section, the data generated by Route Interpolation makes the aircraft run more smoothly. We also combined external data to count the number of flights of passenger and cargo airlines at the current moment. As time increases, you can observe changes in the number of flights of the two types of airlines, as well as the airline with the largest traffic volume. You can also observe that cargo planes are more active in the dark.

4.7.2 *Airports Connection Graph*

In the Find the Main Route section, we found the departure and landing airports of all flights, combined with the aircraft types obtained in the classification, we discovered the airports involved in cargo transportation and the airports only involved in passenger transportation. On this basis, we show the connection between these airports through the Hierarchical Edge Bundling method provided by D3.js. As shown in Figure 7, hover on a airport, you can the cargo and passenger air route that connected to it.

4.7.3 *Transport Volume Comparison Diagram*

We use the frequency of aircraft takeoffs and landings to measure airport traffic. On this basis, in Figure 8 we show

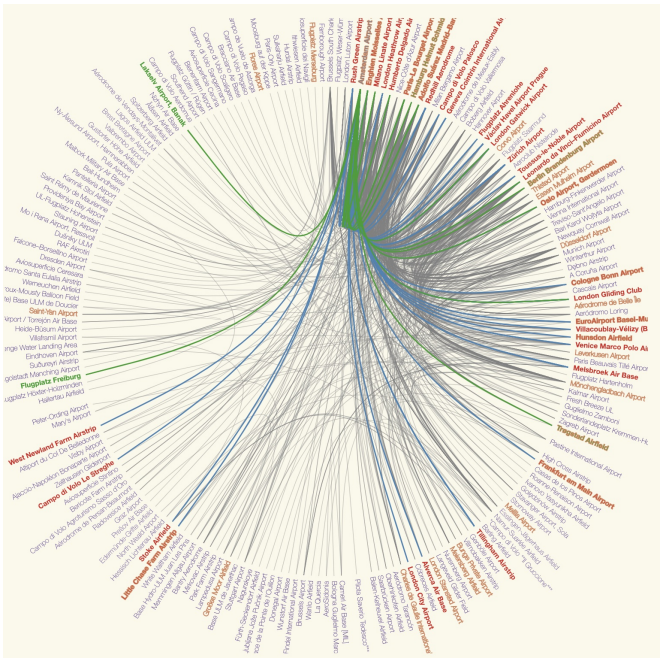

Figure 7: Passenger and cargo airports connection graph

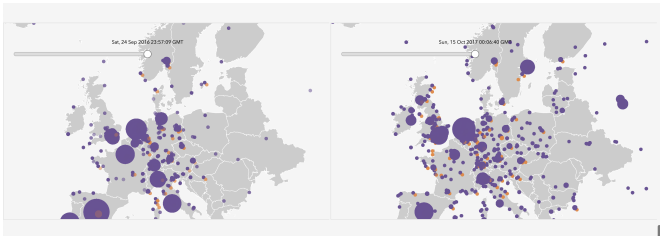

Figure 8: Passenger and cargo volume comparison diagram within two years

how the passenger and cargo volumes of each airport in Europe have changed over time. The picture on the left shows the traffic volume in 2016, and the picture on the right shows the traffic volume in 2017. Therefore, it is easy to compare the passenger volume and the freight volume in the two years.

5. OBSERVATION AND EXPERIMENTATION

5.1 Biggest Operators

For both cargo and passenger aircraft, we respectively counted the five airlines with the highest transportation frequency in 2016 and 2017. Looking at the overall trend, the two types of transport volumes have shown a clear increasing trend. For cargo aircraft, we found that cargo flights usually have a lot of traffic on Sundays, and then the traffic gradually decreases. The following is a specific analysis.

5.1.1 Top 5 Biggest Air Cargo Operators

Figure 9 shows top 5 biggest air cargo airlines in 2 years.

In 2016, the airline with the largest cargo transport volume was Iberia Express (IBS). In the 7-day data, a total of 70 cargo flights belonged to Iberia Express airlines. Followed by European Air Transport Leipzig GmbH (SXS) Airlines, a total of 19 cargo flights. Sun Expre (SXS), (FedEx Express) FDX, and Qatar Airways (QTR) ranked third to fifth, with 18, 11 and 7 cargo flights respectively.

In 2017, Iberia Express (IBS) is still the airline with the largest cargo transport volume, with 83 cargo flights. FedEx Express's (FDX) cargo transport volume has increased significantly, ranking second with 51 flights. The third airline in cargo volume is United Parcel Service (UPS), which has 33 cargo flights in total.

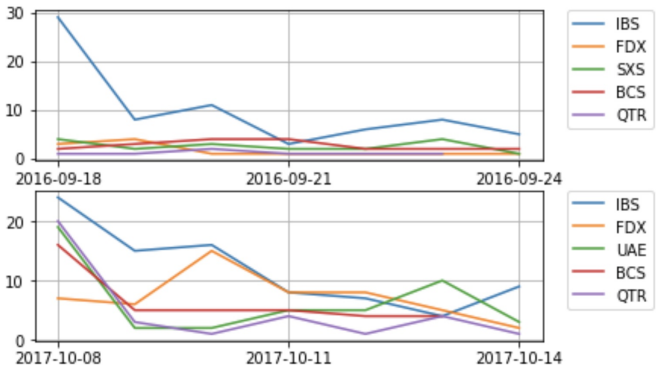

Figure 9: Top 5 Cargo Airlines

5.1.2 Top 5 Biggest Air Passenger Operators

Figure 10 shows top 5 biggest air Passenger operators in 2 years.

EasyJet Airline (EZY) Airlines had the largest passenger traffic volume in 2016, reaching 458 flights in a week. Ryanair (RYR) Airlines ranked second, with 408 flights in a week. Followed by Compagnia Aerea Italiana (AZA), Air France (AFR) and TAP Air Portugal (TAP), there were 317, 282, and 268 flights respectively in the week.

In 2017, in one week's data, Ryanair's (RYR) passenger traffic surpassed EasyJet Airline (EZY), ranking first with 680 flights, and EasyJet Airline (EZY) ranked second with 660 flights. United Airlines (UAL), Southwest Airlines (SWA) and Scandinavian Airlines (SAS) ranked third, fourth and fifth with 341, 325 and 314 flights.

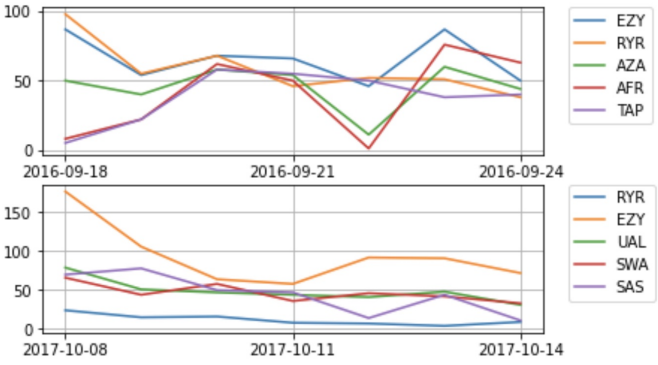

Figure 10: Top 5 Passenger Airlines

We also counted the top ten airlines in these two years, as shown in the Table 6.

Airline	Number of flights
Ryanair Designated Activity Company	1099
Easyjet Airline	1047
Air France	672
British Airways	562
Klm Royal Dutch Airlines	383
Lufthansa	365
Iberia	353
Scandinavian Airlines System	351
United Airlines Inc	319
Southwest Airlines Co	265

Table 6: Top Ten Airlines in 2 years

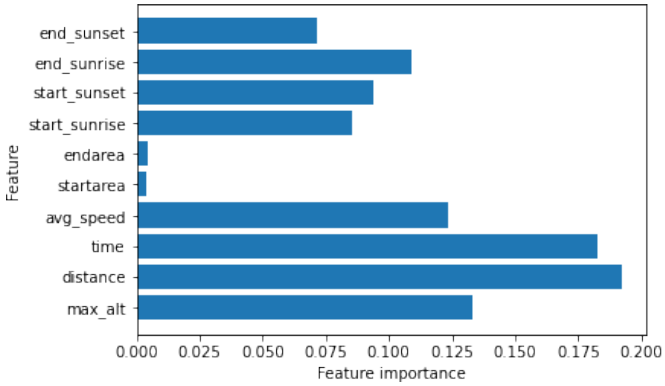

Figure 11: Feature importance

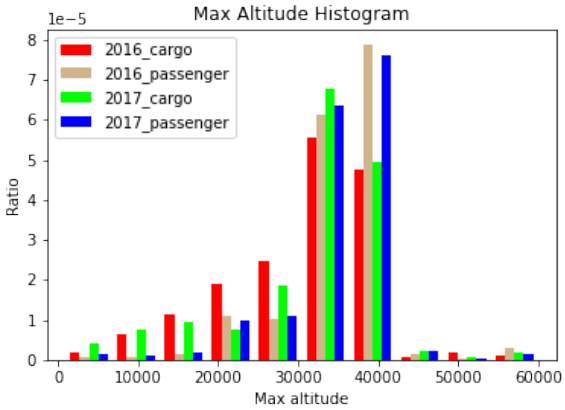

Figure 12:

5.2 Comparsion of the air cargo flights with passenger flights

Figure 11 is the feature importance obtained from the random forest model. We can observed that flying time and absolute distance are the most important feature to separate those two types. It illustrates that these two features are different in most of the cargo and passenger data. This can also be verified by Table 7. Table 7 is the statistical result of the data after classification and outliers elimination. We can figure out that the mean of distance and time of cargo are significantly bigger than passenger no matter in 2016 or 2017. The result is easy to interpret in terms of real life. The minimum flight time may be only one or two hours, due to the transfer, close destination and other reasons. Passenger flights can also take a long time, with direct international flights across the continent or the Atlantic taking more than 15 hours. However, due to factors such as passenger volume and flight cost, there are generally fewer flights in this situation. In addition, the number of routes with short time and short distance is much more than the number of routes with long time. According to our data statistics, the mean of passenger aircraft flight time in 2016 and 2017 was about 4 hours, and the median was about 3 hours, indicating that middle-distance routes accounted for the largest proportion of passenger routes.

As for freight, short distance freight can be handled by car and train, while long distance freight can be carried by sea, which is much cheaper for the same volume. Nearly 99 percent of the tonnage of worldwide trade is shipped. So only the goods with high and critical value are flown, such as electric device like iPhone, laptop and clothing that needs to get to market in time like Zara. Besides, the goods which need to transport under controlled temperature conditions can only secure by air freight. Short distance air freight does not meet the market demand[16]. That's why there are so few passenger flights. According to our statistics, the average flight time of cargo is about 6 and a half hours.

Figure 12 is density histogram of the max altitude distribution. From the histogram, we can figure out that the maximum altitude of passenger and cargo flight is slightly different. The maximum altitude of passenger aircraft is more distributed at about 4000 meters, while the maximum altitude of cargo aircraft is mostly distributed at 3500 meters. Furthermore, because our cargo data may cover small transport aircraft, the maximum altitude of a small amount of data is below 2000 meters.

Besides, according to the Figure 11, the difference between takeoff and landing times and sunrise and sunset also affected the classification. Combined with the analysis of our visualization product, we can conclude that cargo planes are more inclined to fly at night, while passenger planes are more likely to fly during the daytime. These features only have a slight impact on our model, not a decisive one.

5.3 Addition details on cargo flights

We find the outside sources data on the OpenskyNetwork [17] which contains the icao24, aircraft model, aircraft owner information. We also find the cargo payload, capacity, etc. of typical freighters in Wikipedia [18]. For airlines that specialize in cargo transportation usually have obvious signs in their names, such as Cargo or transport. We retrieved all airlines with these signs in their company names on Airlinecodes [9] as an external dataset of cargo airlines. Integrate the outsource data with the parsing data and make a simple statistic, we get the Figure 13 Figure 14 below. Through payload histogram, it can be conclude that our freighter data covered some amount of small cargo transport aircraft and the maximum payload of 200,000kg occupies the largest proportion. We can also find that Boeing 737, Boeing 777 and Airbus A320 are the most popular cargo aircraft model in 2016 and 2017. Furthermore, largest cargo airlines with the type of cargo aircraft shown in Figure 13 are counted. The statistic result is in Table 8. Combined with the previous analysis of the top 5 biggest air cargo operator, we can conclude that large cargo airlines which have

	cargo2016	passenger2016	cargo2017	passenger2017
median_distance	864.4	745.3	1282.5	1107.4
mean_distance	1014.6	951.2	2173.8	1783.2
median_time	12257.1	10221.7	14759.6	9939.5
mean_time	24085.6	17185.9	23431.2	16620.8

Table 7: Comparison of cargo and passenger airline

greater transport volume have more large cargo transport aircraft with considerable carrying capacity.

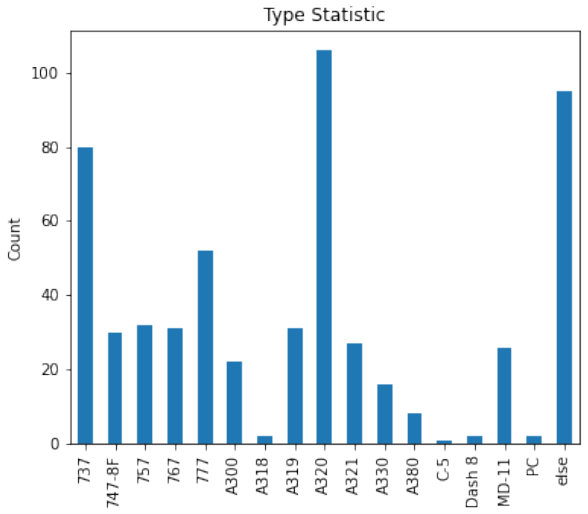

Figure 13: Model Statistic of cargo

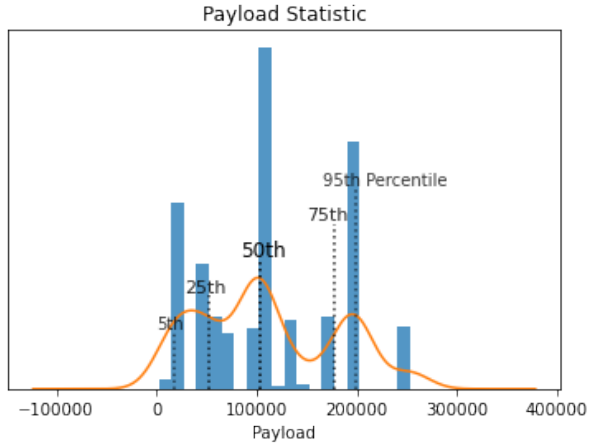

Figure 14: Payload histogram of cargo

5.4 Comparison of major passenger and freight routes

In order to compare the main routes of passenger and cargo flights, and since cargo flights are much less frequent than passenger flights (roughly 6% of all flights), we filtered all airports that contain cargo flights and ran the data on them, as shown in the Table 9. The table is explained as follows: Pairs of airports represents how many pairs of airports

Model	Cargo airline	Amount
737	Sunexpress	15
757	Federal Express Corp	13
777	Qatar Airways	15
A320	Iberia Express	35
MD-11	Federal Express Corp	15

Table 8: The cargo model statistic of the biggest operator

Year	Pairs of airports	Cargo flight	Ratio
2016	91	54	59.3%
2017	235	178	75.7%

Table 9: Percentage of cargo aircraft acting as primary route

(take-off and landing) contain cargo flights, and Cargo flight represents how many pairs of airports have cargo flights as their main route. Given that cargo flights are already relatively small, but also account for more than half of the main routes, we conclude that the main routes of cargo flights are the same as those of passenger flights. This conclusion can also be proved in our visualization products.

5.5 Biggest Airports

Figure 15 shows the airport with the largest cargo volume in the one week sample data of 2016 and 2017. On the whole, the freight volume of the 2017 sample has increased significantly compared to 2016. Adolfo Suárez Madrid–Barajas Airport has ranked first in terms of freight volume for two years. There are still some airports that had very little cargo volume in 2016, but increased significantly in 2017. For example, Lavrentiya Airport did not rank in the top 10 in terms of freight volume in 2016, but it increased significantly in 2017, ranking second. Akhty Airfield and Corvo Airport had very little cargo volume in 2016, but their cargo aircraft in the 2017 sample were both more than 30 flights.

6. CONCLUSION

After parsing the raw message and cleaning the data, we use the track detection algorithm to cluster the data point into a route. Due to the limitation of the number of OpenSky ground sensors, the coverage of some areas is not very good, and due to various interference factors in reality, the extracted track points are intermittent or have a few outliers. To compensate to some extent for the small number of cargo flights, we used a classification model(random forest) to extract the differences between cargo and passenger flights, and then assigned a label (passenger or cargo) to those flights. This gave us a larger sample to study some of the attributes of cargo flights. When comparing the main routes for passenger and cargo traffic, we chose to find the

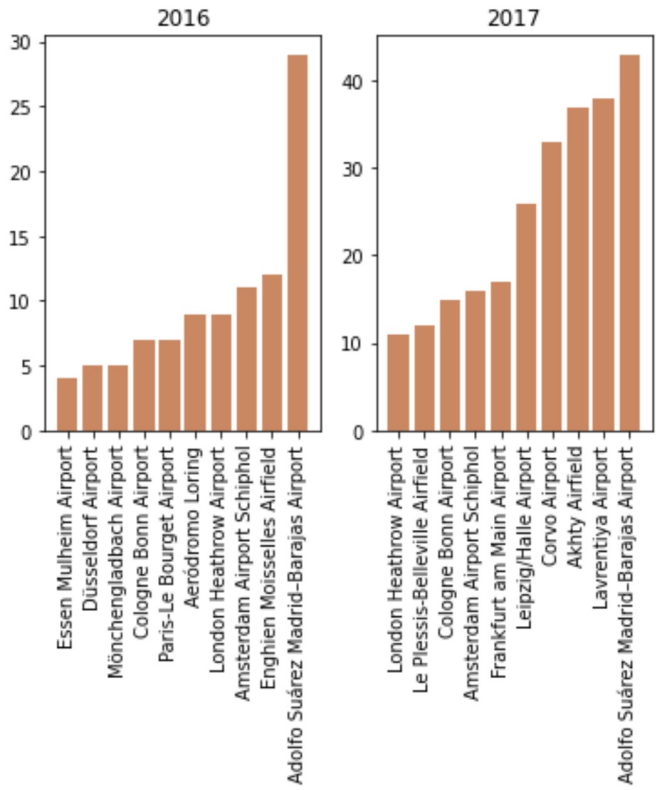

Figure 15: Top 10 airports with the largest cargo volumes in 2016 and 2017

main routes for passenger and cargo traffic together, otherwise, the results after clustering would be more biased. Finally, we use all the processed data to make our data product. We visualized the flight trajectories of two types of aircraft and showed the number of aircraft of airlines. After that, we visualized the connection between airports and their passenger and cargo transport volumes. From the visualization, we see a significant increase in passenger and cargo flights in 2017 compared to 2016.

After previous process, we successfully sharp over 93% of original data. To visualize the route, we construct the route interpolation which eventually reduce the amount of data used for analysis from 800GB to 68GB. After classification, cargo flights account for about 7.5% of the total. According to the feature importance of the classification model and the subsequent data statistics, the average flight distance and time of passenger aircraft are longer than that of cargo aircraft, and cargo aircraft tend to fly at night, while passenger aircraft are opposite. We combined external data to find important characteristics of cargo planes, such as operator and payload, and analyzed them in detail. When looking for the main routes, we found that the routes of passenger and cargo aircraft do not differ significantly and can be considered the same.

7. FUTURE WORK

In our work, there are still many imperfect algorithms and data processing methods, and we will continue to improve after the end of the project. Especially in the trajectory interpolation, we did not deal with the outliers in the original data set, which made the interpolation results of some routes deviate greatly from the correct routes. These interpolation results also affected the subsequent main route clustering.

In the first module of visualization, we display the operation of passenger and cargo aircraft in one year at 1000 times speed, which makes the operation of the aircraft not smooth enough. In the future work, we will expand the playback speed to 100 times, and at the same time, we will face the problem of Javascript program processing large-scale data. We need to read the data in time periods and set the data cache to help the aircraft trajectory display smoothly.

We have also obtained an external data set on the correspondence between the aircraft ICAO and the aircraft model. Through the combination with the airport data, we can further count the aircraft models that take off and land at each airport, and estimate the airport's capacity to carry Maximum passenger capacity and maximum cargo capacity.

8. ACKNOWLEDGEMENT

We would like to thank: Prof Peter Boncz and TA Gabor Szarnyas for teaching the 2021 LSDE course and counseling us on the specifics of the project. During the development of our project, Peter and Gabor gave their immense support and helped us get results of better quality. Thanks to them for their tireless help in solving queries and finding bugs in the code, even at midnight!

9. REFERENCES

[1] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*, 2nd ed. TU Delft OPEN Publishing, 2021.
[2] https://opensky-network.org.
[3] B. L, "Random forests," in *Machine Learning*, 2001, p. 5–32.
[4] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, p. 278–282.
[5] G. D. Amit Y, "Shape quantization and recognition with randomized trees," p. 1545–1588, 1997.
[6] J. S. DeArmon, C. P. Taylor, T. Masek, and C. R. Wanke, "Air route clustering for a queuing network model of the national airspace system," in *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014, p. 2161.
[7] https://github.com/openskynetwork/java-adsb/.
[8] https://ourairports.com/.
[9] https://airlinecodes.info/.
[10] https://en.wikipedia.org/wiki/List_of_passenger_airlines/.
[11] https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-pytl
[12] https://en.wikipedia.org/wiki/F-score.
[13] https://en.wikipedia.org/wiki/Receiver_operating_characteristic.
[14] https://towardsdatascience.com/understanding-random-forest-58381e0602d2/.
[15] https://github.com/d3/d3-geo-projection.
[16] http://www.aircargopedia.com/airlinecargoproductspg.htm/.
[17] https://opensky-network.org/datasets/metadata/.
[18] https://en.wikipedia.org/wiki/Cargo_aircraft/.

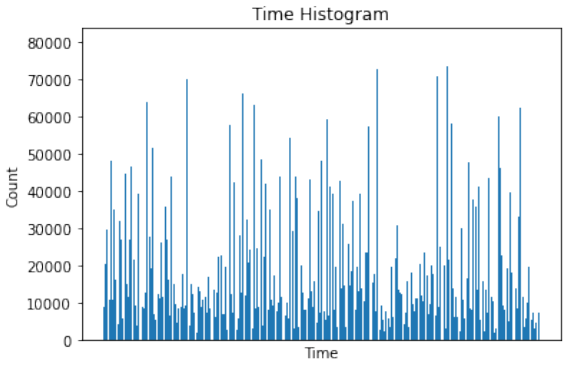

Figure 17: 2016 passenger flight

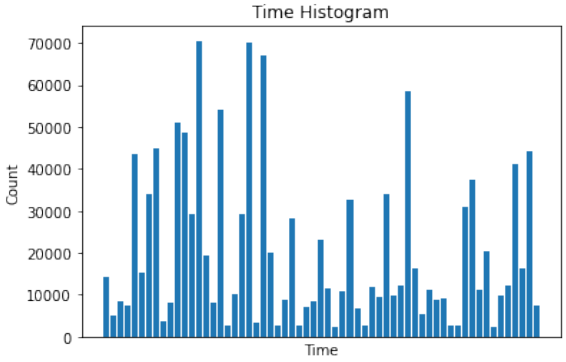

Figure 18: 2016 cargo flight

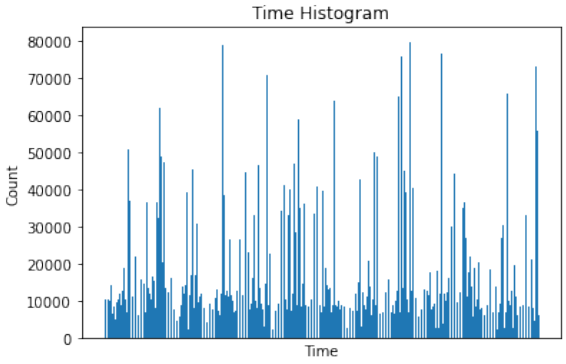

Figure 19: 2017 passenger flight

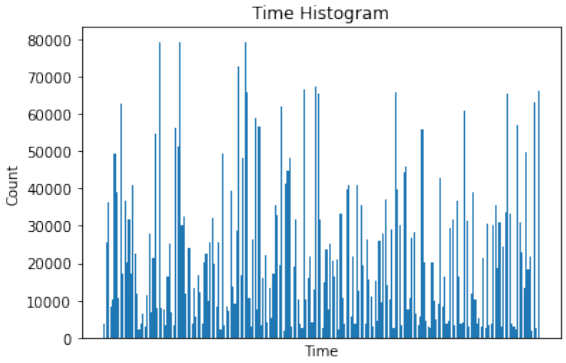

Figure 20: 2017 cargo flight

Project Plan	Haohui, Yongqing, Yiming
Data Extraction	Haohui, Yongqing, Yiming
External Data Collection	Haohui, Yongqing, Yiming
Decoding ADS-B	Haohui, Yongqing
Data Processing	Haohui, Yongqing, Yiming
Preliminary Classification	Haohui
Advanced Classification	Haohui
Main Route Clustering	Yongqing
Data Project Generation	Yongqing, Yiming
Experimentation & analysis	Haohui, Yiming
Data Visualization	Yiming

Table 10: Code Contribution

Abstract	Yiming
Introduction	Yiming, Yongqing
Related Work	Yongqing, Haohui
Project Setup	Haohui, Yongqing
Observation & Experimentation	Yiming, Haohui, Yongqing
Conclusion	Yiming, Haohui, Yongqing
Future Work	Yiming

Table 11: Report Contribution

APPENDIX

A. CONTRIBUTION

Table 10 and Table 11 show our work contribution.

B. VISUALIZATION ON DBFS

Our visualization can be found on DFBS at dbfs:/FileStore/group16/Group16_AirCargo_visualisations-1.zip

C. STATISTICS SUPPLEMENT

Figure 17, 18, 19, 20 are the histogram of flying time of all cargo and passenger flights in 2016 and 2017.

Figure 16 is the histogram of flying maximum range of all cargo aircraft.

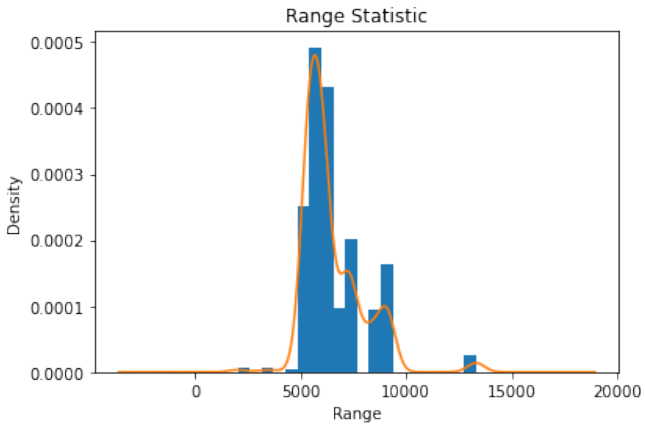

Figure 16: Cargo maximum range statistic