Spatial Analysis of ADS-B Messages to Identify Rescue Helicopters Using Spark

Richard Gankema
Vrije Universiteit
De Boelelaan 1105
Amsterdam, Netherlands
r.gankema@student.vu.nl

Oliver Becher
Vrije Universiteit
De Boelelaan 1105
Amsterdam, Netherlands
o.l.becher@student.vu.nl

ABSTRACT

Due to growing air traffic, a new air traffic management system, ADS-B, is about to be fully adapted by all kinds of aircraft. Also helicopters start to use ADS-B for communication. OpenSky Network provides a data set containing a large amount of ADS-B messages from September 2015, which covers selected areas of Europe. The purpose of this paper is to identify helicopters and especially rescue helicopters in that data set. The presented classification algorithm coarsely filters out aircraft that surely are not helicopters, and analyzes the flights of the remaining aircraft to identify helicopters. The classifier focuses on typical helicopter flight patterns, especially during take off and landing sessions, which is particularly challenging because signals are often lost during these phases in flight. This gives a remaining set of about 62 helicopters which were compared to hospitals to find rescue helicopters. Unfortunately, no rescue helicopters could be found using this data set and our algorithms. However, the research remains an interesting exercise in dealing with large quantities of spare ADS-B data. The outcome is a visualization of all helicopters in Europe that were found, and the areas their flights covered during the time span of the data set.

Keywords

Large Scale Spatial Analysis; Spark; ADS-B Analysis; Rescue Choppers

1. INTRODUCTION

With air traffic ever growing, new ways of managing it are required. One of the new tools in doing this is the ADS-B protocol, short for Automatic Dependent Surveillance. The main idea behind it is that aircraft locate themselves using GPS, and broadcast this location to other aircraft and control centers. This allows the location of aircraft to be detected more accurately than with radar. Unlike radar, ADS-B also allows a receiver to gather much more information than merely the position of an aircraft. Among the broadcasted information are an aircraft's speed, heading, rate of climb, identification, and more.

ADS-B will be mandatory for aircraft in the European airspace by 2020 [7]. Also, ADS-B messages are not encrypted, which means anyone with a sensor can record messages sent in its vicinity, and decode them. This makes it possible to analyze large amounts of detailed aircraft flight data for a wide range of research purposes.

The goal of this paper is to find out if ADS-B messages can be used to identify rescue helicopters in Europe, determine where they are stationed and which areas are (not) covered by these helicopters. Information like this could provide an international overview of resources that can be shared or should be relocated elsewhere. This is relevant in a continent like Europe where many countries are collaborating.

The remainder of this paper is structured as follows. In section 2 related work is discussed, while in section 4 the structure and content of the data that we work with is explained in detail. Section 3 discusses the research questions, section 4 describes the data that we have used, section 5 describes the steps that were taken to identify rescue helicopters, and is followed by section 6 which describes the experiments that were done for this research. Finally, we conclude the paper in section 7.

2. RELATED WORK

Our research builds on previous work related to ADS-B, large scale data engineering, and the combination of the two. This section briefly summarizes related research.

2.1 ADS-B

ADS-B is a protocol meant to replace surveillance radar technologies [5]. The main philosophy behind ADS-B is that aircraft locate their own position using GPS, and then broadcast this position to other aircraft and surveillance stations. This allows for more accurate positioning [4] than radar. ADS-B messages are unencrypted, and can in carry much more information than just location, such as speed, altitude or identification.

2.1.1 OpenSky

Because ADS-B messages are unencrypted, anyone with a sensor can record them, and analyze them. OpenSky [5] is a network that does just this. The goal of OpenSky is to collect ADS-B messages so that they and others can use them for scientific research. The network is not just run by researchers, but also by volunteers. Anyone with a compatible sensor can contribute, thereby increasing the size of the total network.

2.1.2 Flight Phase Identification

In [6], Sun et al. show that analysis of a large number of ADS-B messages is feasible using a combination of machine learning methods and fuzzy logic. The goal of their research was to identify the various phases of a flight (lift-off, cruise, landing). Particularly relevant to our research is how they dealt with the often incomplete data that ADS-B messages

provide, and how they cluster the extracted data into flights.

The stream of ADS-B messages is converted into a stream of flight data, where data points often have missing values for various fields (such as position, altitude or velocity). To manage the large amount of incomplete data, a NoSQL database is used rather than a traditional relational database. The resulting flight data is clustered into flights, using the DBSCAN and BIRCH clustering algorithms. Finally, fuzzy logic is applied to determine the different phases of each flight.

Although the ultimate goal of the research is different, Sun et al. faced many of the same challenges that are met when identifying helicopters in a large set of ADS-B data. To identify helicopters, the first crucial step is to process the raw data and cluster the resulting flight data into flights, analog to [6].

2.2 Spark

For large scale data analysis, the use of a computer cluster is usually required, typically using Hadoop. Many different paradigms for working on these clusters have been proposed and are currently being used. One of these is Spark [8]. It differentiates itself from MapReduce and its variants in that it offers a less rigid paradigm, without giving up scalability and performance. In fact, because it tries to do as much work in memory rather than on disk, it is often faster than MapReduce.

The core concept of Spark is that of resilient distributed data sets (RDDs). An RDD is an abstraction for a read-only in-memory collection, distributed over a cluster, that can be accessed as if it were in one place. RDDs are called resilient because they can be rebuilt in case a node in a cluster fails. RDDs support a wide range of parallel operations that can be chained to transform one set of data into another, in a fashion that is often more intuitive than MapReduce.

3. RESEARCH QUESTIONS

The goal of this research is to identify rescue helicopters and determine their coverage by analyzing a large set of ADS-B data. The following research questions arise:

1. Where are rescue helicopters located within the range of our data?
2. What is the effective coverage of those rescue helicopters?

To answer the first question we will first need to identify helicopters in the data set, then determine which of those are rescue helicopters, and finally determine where they are stationed. When the first question is answered, we need to determine where the rescue helicopters fly from where they are stationed.

In order to do this, specific software to work with large scale data is required. For this project, Spark is chosen, for its flexible paradigm of transforming RDDs and its relatively high performance. The results are presented using a web based visualization that shows the location of helicopters, and their flights.

4. DATA

This research is done on three different sets of data. The first is a set of mode S (in particular ADS-B) messages collected by sensors from OpenSky Network [5], the second and third are locations of helipads and hospitals extracted from OpenStreetMap [2].

Field	Type
Sensor type	string
Sensor latitude	double
Sensor longitude	double
Sensor altitude	double
Time at server	double
Time at sensor	double
Raw message	string
Sensor serial number	int
RSSI packet	double
RSSI preamble	double
SNR	double
Confidence	double

Table 1: Schema of the OpenSky data set

4.1 OpenSky Network Data

The main set of data is a collection of ADS-B messages broadcasted by aircraft and recorded by sensors scattered across Europe. The set spans the month of September in 2015, and is about 200GB in size. The data is compressed and stored in the avro format. Table 1 shows the schema of data in this set. The raw message is the ADS-B message encoded as a binary string. The other fields are mostly metadata about the sensor and the time of recording.

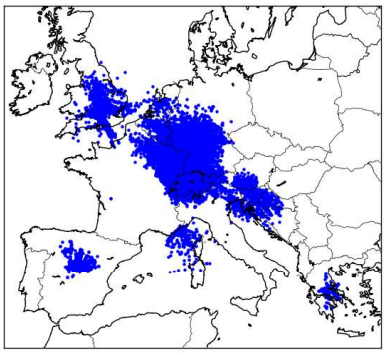


Figure 1: Scatter plot of a sample of recorded positions

The data set consists of a total of about 2.3 billion messages, from about 27 thousand different vehicles. As can be seen in figure 1, the coverage of the data is rather limited. This has several consequences. First, it will not be possible to determine the coverage of rescue helicopters for all of Europe, but only those locations that are actually covered by the data set. Second, only partial flights can be constructed from the data for aircraft that fly outside the boundaries of the data coverage. Our methods will have to be able to cope with this.

4.1.1 Types of messages

Technically speaking, the messages in data set are not strictly ADS-B messages, but mode S messages, of which

ADS-B is a subset. In the remainder of the paper we will use the term ADS-B for all mode S messages, as the overwhelming majority are ADS-B messages.

Mode S (and ADS-B) messages come in different types. The ones relevant to our research are briefly discussed below. Other messages include, among others, emergency messages and status messages.

1. ADS-B aircraft identification: reports general information about the aircraft, such as the type of aircraft
2. ADS-B airborne position: reports the position and the altitude of the aircraft while it is in the air
3. ADS-B surface position: reports the position of the aircraft while it is on the ground
4. ADS-B airborne velocity: reports the velocity of the aircraft while it is in the air, as well as its heading and its rate of climb
5. (Comm-B) altitude reply: reports the altitude of an aircraft

Table 2 shows the distribution of message types in the data set. As can be seen, most messages by far are either airborne velocity messages or airborne position messages. The remaining messages are mostly identification messages, Comm-B altitude replies and ADS-B messages that the decoder was not able to decode into anything meaningful. The relatively high amount of identification messages makes it so that many aircraft can already be identified using their category descriptions. The high amount of airborne messages means that there should be enough flight data in the covered areas to model flights for the remaining aircraft.

Type of message	%
Airborne position	42.25
Airborne velocity	42.25
Aircraft identification	5.05
Comm B altitude reply	2.77
Aircraft status message	1.16
Comm B identification reply	0.62
Surface position	0.59
Airborne airspeed	0.22
Emergency	0.15
Other	4.94

Table 2: Message types in the ADS-B data

4.2 OpenStreetMap Data

OpenStreepMap provides open source map data with very detailed information which can be used for the creation of special maps. It is strongly enriched by users. To put helicopter position messages into context, this map data is a valuable source. We obtained the map data from Europe which is about 19GB in a compressed format. The map data consists of nodes, ways and relations.

Nodes define landmarks for any kind of object at a specific position. Ways define boundaries for a specific object and relations as a poly-line and relations are a multi-purpose structure to combine ways and nodes with restrictions. Each of these entities can be enriched with tags which can be any kind of key-value pair.

We filtered the map data on ways and nodes to create two sets, one containing locations of hospitals and one containing all helipad locations. These are shown in figure 2a and 2b.

5. PROJECT SETUP

5.1 Analyzing Data

To better understand the data that we were working with, a Spark job was run on the entire data set that analyzed it. The following information was then extracted:

1. The number of recorded messages
2. The number of aircraft
3. The distribution of messages over time
4. The type of messages and their distribution
5. A sample of reported positions
6. A sample of reported altitudes
7. A sample of reported velocities

Some aircraft send identification messages, with which they report the category of aircraft that they belong to. Helicopters is one of those categories. Because not all aircraft identify themselves these messages are not sufficient for identifying all helicopters in the data set. However, they still provide a valuable source of information for identifying other helicopters. Using these identification messages, the following information was also acquired:

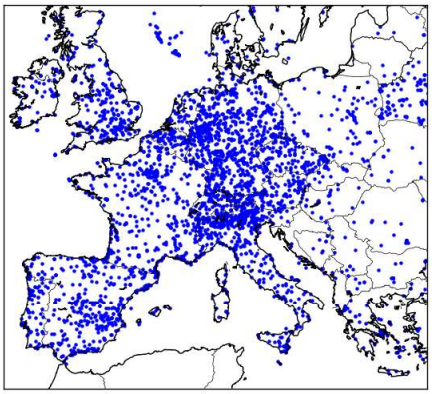
8. Reported altitudes of helicopters
9. Reported velocities of helicopters

The results of 1 to 5 are discussed in section 4. Figure 3a and 3b show the distribution of the altitudes and velocities reported by all aircraft, whereas figure 3c and 3d show the same for just the helicopters. As can be seen, most aircraft fly between 10km and 12km, at a speed between 200m/s and 270m/s. Helicopters however rarely fly higher than 3km or faster than 90m/s. This suggests many aircraft can already be excluded by looking at how high and fast they fly.

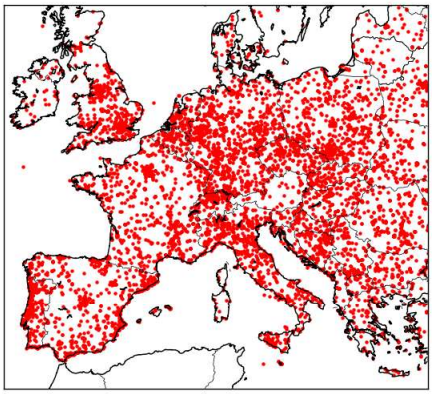
5.2 Initial Filtering of Data

Because we are only interested in helicopters, the majority of the data is irrelevant to us. Therefore, the first step in finding rescue helicopters consists of excluding all ADS-B messages from aircraft that are definitely not helicopters. A Spark job was run that does this in the following way:

1. Messages are filtered so that only those that are of use for finding helicopters remain. This means that among others ADS-B emergency messages are excluded. Because most messages are actually relevant messages, this is only a minor improvement regarding the amount of messages.
2. Messages are grouped by aircraft, and groups are filtered out when one of the following conditions is true:

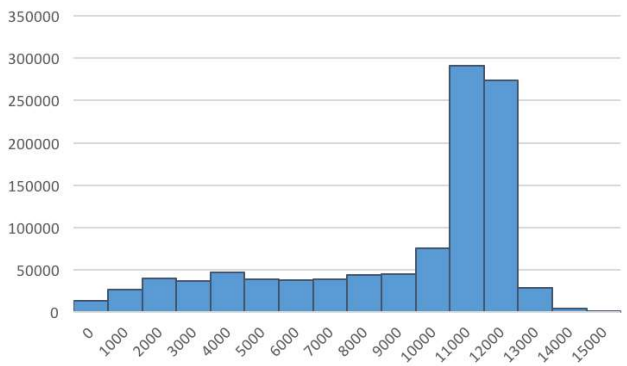


(a) Locations of Helipads

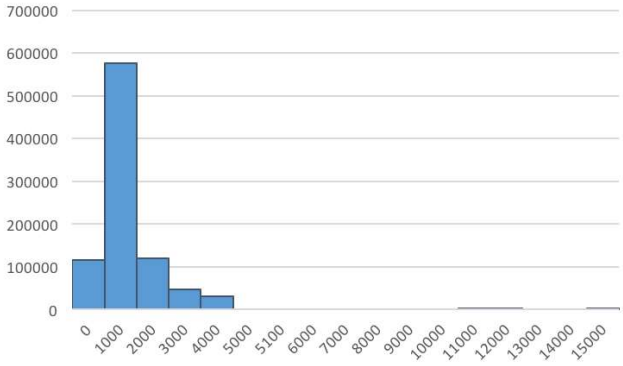


(b) Locations of Hospitals

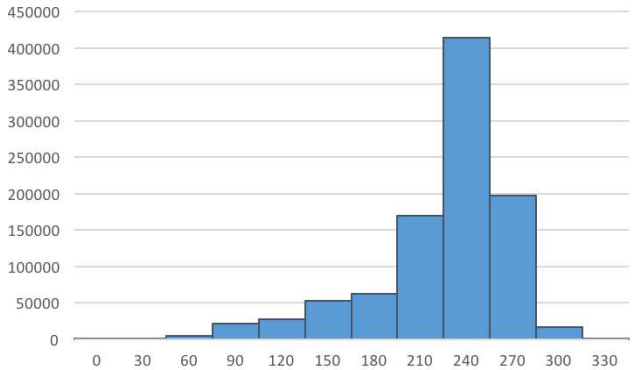
Figure 2: OpenStreetMap Data



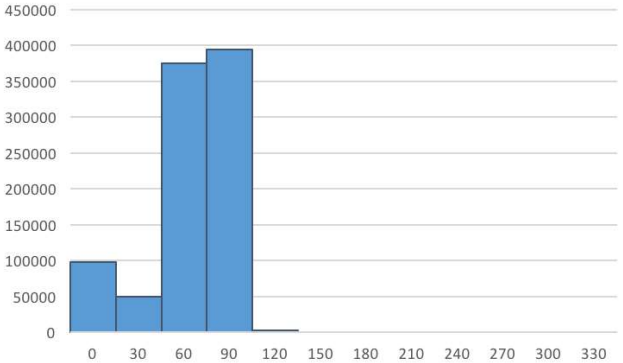
(a) Sample of all aircraft altitudes (m)



(b) Rotorcraft altitudes (m)



(c) Sample of all aircraft velocities (m/s)



(d) Rotorcraft velocities (m/s)

Figure 3: Histograms

- The aircraft sent a message reporting an altitude above 3500 meters
- The aircraft sent a message reporting a speed above 100 meters per second
- The aircraft sent an identification message reporting a category other than 'Rotorcraft' or 'Unidentified'
- The aircraft sent a message of the military squitter type, meaning the aircraft is a military aircraft

3. The groups are flattened, and each data point is converted into CSV format containing the raw ADS-B message, sensor latitude and longitude and a timestamp. Other information such as the identifier of the sensor that recorded the message are left out, because they are irrelevant for identifying helicopters. This way, the data set becomes even smaller.

The resulting data set comprises of about 2.5 million messages from 1075 aircraft. The size is 160MB, compared to over 200GB for the original data. Not only does this make future analysis far more efficient, it also doubles as an important first step in identifying helicopters.

5.3 Finding Flights

Because not all aircraft report whether they are a helicopter or not, flight data for the remaining aircraft has to be analyzed to identify helicopters. Also, the flights of all helicopters are analyzed to identify rescue helicopters. This section explains how individual flights are identified, and how they are analyzed to determine (rescue) helicopters.

5.3.1 *Extracting Flight Data*

The first step towards identifying flights is to convert the stream of sensor data into a stream of flight data containing the aircraft's icao, timestamp, position, altitude, velocity, rate of climb and heading. Each point of flight data is represented using a `FlightDatum` object, which has the schema specified in 3. Note that a flight data point does not necessarily have data for all fields. This is because there is no message that contains all the required data, and not all aircraft send all types of messages. The stream is created as follows:

1. The set of messages are filtered so that only messages containing flight data are left. These types of messages include airborne velocity, airborne position, airspeed, surface position, Comm-B altitude and altitude messages.

2. All messages are grouped by aircraft, and each group is then sorted on time. Sorting on time is necessary because the position of an aircraft is encoded using two position messages.

3. Each group of messages is mapped to a group of `FlightDatum` objects. This results in a stream of around one or two `FlightDatum` objects per second.

4. Each group of `FlightDatum` objects is clustered into 5 second windows, and all `FlightDatum` objects in each window is then merged into single `FlightDatum` objects. This leaves us with more complete individual `FlightDatum` objects, and overall less flight data points. One point of flight data for every 5 seconds should still be sufficient to analyze a flight.

Field	Type	Unit
ICAO address	string	-
Timestamp	double	s
Latitude	double	deg
longitude	double	deg
Altitude	double	m
Heading	double	deg
Velocity	double	m/s
Rate of climb	double	m/s

Table 3: Schema of the `FlightDatum` class

5.3.2 *Clustering*

After creating streams of flight data for every aircraft, flight data are clustered into individual (partial) flights. This is done in two steps, the first of which is looking at the time between consecutive flight data points. For each aircraft, the flight data are iterated upon in order of time. A list of previously visited flight data points is kept, as well as the time of the last `FlightDatum`. Each time that the time between the current `FlightDatum` and the last is one more than 20 minutes, all the previous flight data points are grouped into a single flight, and the list of previous visited flight data is cleared.

Some aircraft in the data set did not report any location data. Therefore, flights were generated that did not contain any location data. These are not of any use to us, and were therefore discarded. Also any (partial) flights that lasted for a very short time were filtered out, as they proved to be useless for any meaningful analysis. These flights belong to aircraft that entered the area covered by the sensor for only a very short time.

5.3.3 *Determining Landings*

The second step in determining flights is splitting the previously generated flights on moments that the aircraft landed. This is primarily done by looking at the altitude during the flight. If it drops under 30 meters, the aircraft is considered to have landed. The reason 0 was not chosen, is because sensors may not have line of sight on aircraft at low altitudes. Most of the times the landing would then have already been correctly identified by splitting flight data on time, but may fail in case a helicopter would land for a short time, and then quickly take off again.

Not all aircraft report altitude data, so for those aircraft a different method is required. Rather than looking at altitude of a flight, the position of the aircraft is considered. For each minute in time, the central position of all reported positions during that minute is calculated. If the distance between the central position and any of the reported positions is less than 8 meters (the effective maximum error of GPS equals 7.8 at 95 % confidence level [3]) for the entire minute, the aircraft is deemed to be standing still, and thus most likely on the ground. In rare cases it may be a helicopter hovering above ground, but looking at all positions for an entire minute rather than just two data points should make this an unlikely event.

The process resulted in 1390 flights, for 179 aircraft. Note that we are left with about 6 times less aircraft than after the initial filtering of sensor data. This has to do with how scattered the area covered by the sensors is. Many aircraft only remain in this area for a short time, resulting in very short flights within that area, which are thus filtered out. Also, as mention before, some aircraft don't report location data at all.

5.4 Classifying Aircraft

Now that flights have been generated for each aircraft, we can try to identify whether an aircraft is a helicopter or not, and whether a helicopter is a rescue helicopter or not. Several characteristics of a flight typical for helicopters were identified:

- Helicopters may hold position while airborne (hover), while airplanes cannot
- Helicopters may fly at lower speeds than airplanes
- Helicopters may ascend or descend at higher angles than airplanes
- Helicopters generally land at helipads, while airplanes land at airstrips

To check whether an aircraft has been hovering during a flight, the flight data of a flight is clustered by 30 second windows. If the positions during this time span did not deviate more than 10 meter, and the aircraft had been airborne, it is determined that the aircraft was hovering. To check whether an aircraft has been flying at low speed, velocity and altitude data is checked. If altitude is above 0 meters, and velocity below 15 m/s, it is marked as low speed flying. 15 m/s was chosen as it is well below the stall speed of a light airplane like the Cessna 172 (around 20m /s) [1]. At first sight determining whether an airplane is hovering may seem unnecessary since the ground speed of an aircraft is also taken into account. However, for flights that do not contain velocity data this can be a useful backup method.

Checking whether an aircraft ascends or descends like a helicopter, the angle between rate of climb and velocity is calculated. When this angle is higher than 50 degrees, we're likely looking at a helicopter. Finally, to check whether an aircraft landed at a helipad, the location of the start and end of each flight is compared to known locations of helipads. If the aircraft was below 100 meters, and within 300 meters, it is marked as landing at a helipad. The extra margin in altitude and distance is used in case the aircraft loses line of sight with the sensor because of its low altitude.

Of the 179 aircraft we generated flights for, 117 could not be identified as helicopters by their identification messages. Their 445 flights are analyzed for helicopter characteristics. If at least two of the conditions hold true on one or two flights of the aircraft, that aircraft is classified as a helicopter.

No helicopter behavior was detected for any of the 117 aircraft. Note that although we have not managed to detect any helicopters among the unidentified aircraft, this does not mean the classification does not work. As will be discussed in section 6.1, the classification method has been evaluated, and proven to be able to detect helicopters. However, the classifier is not able to detect all helicopters and there were relatively few flights to analyze for each unidentified aircraft, so it's possible some helicopters were not identified.

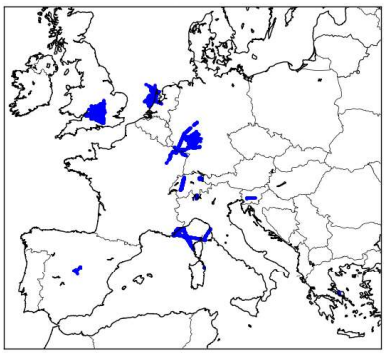


Figure 4: Reported positions from identified helicopters.

5.4.1 *Identifying Rescue Helicopters*

After identifying helicopters, either by looking at identification messages or their flight patterns, rescue helicopters can be found. This is done using a method similar to determining whether aircraft land at helipads. However, instead of comparing the start and end locations of flights to those of helipads, they are compared to known locations of hospitals in Europe. If the helicopter is detected to land at one of the hospitals, it is classified as a rescue helicopter.

In the set of 62 helicopters, no rescue helicopters could be found. This can have a number of reasons. First of all, as the coverage of the data set is rather poor, many flights are not complete. Partial flights are not a problem per se, but we do need data on either landing or lift-off so we can determine where a helicopter is stationed. Unfortunately, many flights started and ended at rather high altitudes, meaning that we cannot determine this. Another problem is the fact that 62 helicopters is in fact not a very large number of aircraft, and it's thus not unreasonable that there are no rescue helicopters among them in the first place.

5.5 Visualization

As a final result we produced an interactive visualization of the helicopters that were found. This includes first a map which provides an overview of all helicopter flights and an indicator of airspace coverage and, secondly, a detailed visualization for flight trajectories per helicopter, including an altitude per distance plot per flight. This was mainly done by defining templates with Python and injecting those into a JavaScript library called Leaflet to produce maps with the helicopter data. The underlying map is based on OpenStreetMap with an overlay from CartoDB.

6. EXPERIMENTS

Apart from writing Spark jobs to find rescue helicopters, an experiment was performed. The goal of this experiment was twofold: first to find the best parameters for the helicopter classification algorithm, and finally to evaluate this algorithm.

6.1 Flight Classification Evaluation

To evaluate how well the method for helicopter classification works, a separate Spark job was written that classifies

two sets of aircraft: one that are known to be helicopters, and one that are known to be anything but helicopters or unidentified aircraft. These test sets were created by looking at identification messages sent by aircraft. The classification method was then applied to both sets, and the amount of (false) positives and (false) negatives were recorded.

	Positive	Negative
True	12 (20.7%)	249 (98.8%)
False	3 (1.2%)	46 (79.3%)

Table 4: Results of the classifier evaluation

Table 4 shows the results of the evaluation. As can be seen, only about 20% of all helicopters are classified as helicopters, while about 1% of non-helicopters are marked as helicopters. This can perhaps be improved by using better parameters for classifying helicopters, although as mentioned the evaluation method has already in fact been used in finding optimal parameters. Another possible reason for the poor results is the low coverage of the ADS-B data. Lift-off and landing is missing for many of the flights, which means that a lot of aircraft cannot be detected at many of the known helipads. It also makes detecting high angle ascends or descends less likely.

Note however that in fact more than 20% of the helicopters are eventually identified, as this method is only used on those aircraft that do not send identification messages. Helicopters that broadcast a 'Rotorcraft' category description are always correctly identified as helicopters.

7. CONCLUSION

With our setup, we were not able to find any rescue helicopters in OpenSky's September 2015 data set. This means we cannot answer our two research questions. Although the data set contains messages for 27,000 aircraft, only 179 of them potentially were potentially helicopters and had useful flight data. 62 identified themselves as helicopters, while the remaining did not broadcast their vehicle type. Although we have created an algorithm that detects helicopters with moderate success, none of these could be classified as helicopters. This low number of helicopters is one likely cause for the fact that no rescue helicopters could be identified.

So the first question that should be answered is why the number of identified helicopters is so low relative to other aircraft. One explanation is that helicopters generally have a much shorter range than airliners. An aircraft flying from Paris to Singapore will pass the area of OpenSky's sensor network, even though that aircraft is not stationed within the same area. This also explains why already so many aircraft are filtered out during the first filtering step described in 5.2. Another possibility is that in 2015 planes were perhaps outfitted with ADS-B transponders relatively often compared to helicopters. Further research is required to verify whether this is the case.

7.1 Future Work

There are a few suggestions for future work. The first would be to work with a more recent set of ADS-B data that has better coverage over Europe. Denser data makes our classification methods more reliable, and allows for detection of helicopters that are stationed in locations that are not covered by the 2015 data.

Another obvious improvement could be done on the classification of helicopters. There may be more features of a helicopter flight that distinguish it from one of an airplane. One that we can think of is the cruising speed and altitude. Due to lack of time we were not able to implement this properly, but it might yield better results for classifying helicopters.

Similarly, more could be done to detect rescue helicopters. In this research we only look at the lift-off and landing location of helicopters, and compare it to hospital locations. The problem is that this does not work well when sensors don't have line of sight on the area around those hospitals. There may be other features that distinguish a regular helicopter from rescue helicopters, such as their call sign or flying patterns.

8. REFERENCES

[1] Cessna 172P.

[2] OpenStreetMap.

[3] Global positioning system standard positioning service performance standard, 9 2008. Accessible online at http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.

[4] A. Smith, R. Cassell, T. Breen, R. Hulstrom, and C. Evers. Methods to provide system-wide ads-b back-up, validation and security. In *2006 ieee/aiaa 25TH Digital Avionics Systems Conference*, pages 1–7. IEEE, 2006.

[5] M. Strohmeier, I. Martinovic, M. Fuchs, M. Schäfer, and V. Lenders. Opensky: A swiss army knife for air traffic security research. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pages 4A1–1. IEEE, 2015.

[6] J. Sun, J. Ellerbroek, and J. Hoekstra. Large-scale flight phase identification from ads-b data using machine learning methods. In *7th International Conference on Research in Air Transportation*, 2016.

[7] M. Thurber. Europe delays ads-b out equipment mandate, 2014.

[8] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. *HotCloud*, 10:10–10, 2010.