Anomalous aircraft trajectory detection using statistical analysis and machine learning

Group D05 - Tilen Potočnik, Bryan Quadras, Matthijs Slobbe, Servaas Clerckx, Yoni Verhulst, Koen Smit, Pieter van de Ven
AE2221-I, TU Delft

Abstract

Automatic Dependent Surveillance–Broadcast (ADS-B) is an aircraft surveillance technology that can be used to identify flights that show anomalous (not normal) behavior. Recognizing anomalous flights is important because there is potential to make air travel more sustainable if the causes can be identified. Anomaly detection of such data is commonly done using computationally expensive machine learning techniques. This paper explores how anomaly detection and prediction can be done with a heuristic approach, through comparing the results to those achieved using neural networks. The heuristic methods include analysis of flight time and runway approach characteristics. Due to large discrepancies in the number of anomalous flights, as well as a lack of overlap in comparison with the machine learning output, the results suggest that the statistical method of analyzing flight trajectories is inaccurate and hence not a valid approach for detecting anomalous flights. For future research, the group recommends that more sophisticated statistical methods are studied. This may include using more types of data, such as velocity and weather. These could then be used alongside well-known, reputable machine learning techniques for more conclusive identification of anomalies in ADS-B data.

I. Introduction

Automatic Dependent Surveillance-Broadcast (ADS-B) data is the main format of aircraft flight information transfer, as an ADS-B transmitter is required on all commercial aircraft as of 2020 [1, 2]. The data provided through ADS-B signals contains the necessary information to track and analyze an aircraft's flight path. With an ever-growing air transport industry [3], making the skies more sustainable is of prime importance. Anomalous flights have flight paths that deviate from the normal flight path, which could be caused by a go-around or holding pattern for example. Naturally, we want to prevent this type of behavior and thus it is important to detect and eliminate these anomalies using data from ADS-B. Since machine learning is known to be able to deal with significantly large amounts of data accurately, and the number of flights occurring at a given time is substantial, machine learning is often looked to [4, 5].

However, machine learning requires extensive programming and high computing power, which is not accessible to all researchers. An alternative possibility for analyzing the ADS-B data is using a more heuristic approach with statistics. Such a process is much more intuitive and can be used with lower computing power, but the accuracy of the results is not guaranteed.

This report aims to determine the accuracy of a heuristic approach to detect anomalies in comparison to a machine learning approach. Four methods to analyze the data have been explored. These include: anomaly detection using the amount of flight data points; anomaly detection during the final approach; using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6] and using a convolutional neural network. The set of anomalous flight trajectories produced by the heuristic approaches can be validated using the machine learning approaches, which are used as a benchmark for the accuracy. If the anomalous flights detected using machine learning closely match the anomalous flights detected by the heuristic methods, the detection accuracy level is high, and the heuristic approaches are valid for anomaly detection.

II. Methods

In this section, the analyzed data set and the methods of analysis will be discussed. Firstly, in subsection A, the used data set and the approach of processing are presented. subsection B provides the background on the two heuristic approaches and lastly, subsection C gives the information on anomaly detection using machine learning.

A. Processing the Data

The data used by the team consists of decoded ADS-B data of flights arriving at the Zürich airport, provided in the *.parquet* format. The data spans the time frame from October 1st to November 30th, 2019. During this time, the amount of data points collected is approximately 18 million.

One aspect of the data, the longitude and latitude of the aircraft, is presented in its unprocessed form in Figure 1. The figure presents the arrivals part of the data set composed of 19,480 flights. Due to the amount of data and the seemingly chaotic nature of it, it needs to be processed, before it can be analyzed and effectively visualized.

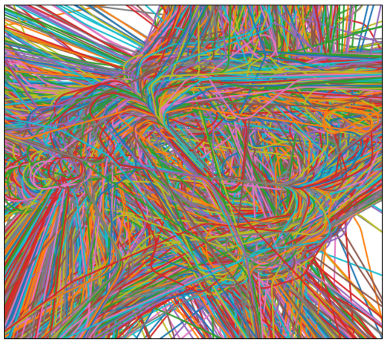

Figure 1: Plotted trajectories of the unprocessed arrival data at the Zürich airport.

Different methods, that use various aspects of the data, require different pre-processing. For methods that only use the longitude and latitude, the data only needed to be separated into sets for the runways. The timestamps of each data point were also used. However for methods using altitude data, the unprocessed data needed to be filtered to eliminate corrupted data points and to refine the data by removing as much noise as possible. This processing was performed using the *traffic* library [7, 8], which is compatible with Python and was developed specifically for air traffic analysis using ADS-B data. The library passes the data through a median filter and clears out any data points with unexpected values [7]. After this processing was performed, other libraries were used for each different research approach and will be detailed later.

B. Heuristic Anomaly Detection

The following methods of anomaly detection rely on the usage of statistical data. These methods are meant to require low amounts of processing power, while still providing satisfactory levels of accuracy.

1. *Detection using number of data points*

The first approach was to use the number of data points provided per flight to detect anomalies. The ADS-B transfers data at a set frequency, therefore two equally long flights have the same number of data points. Normally, flights around the airport should be relatively straight. Therefore, an increased amount of data points for a flight would mean an anomalous behavior, such as a hold or go-around. To our knowledge, this method has not been used before, hence we wished to investigate the accuracy of this method.

To be able to detect anomalies using statistical analysis, a public code using Python was set up [9]. Firstly, the arrival data set was extracted using the *traffic* library. Then, the data was filtered such that the Flight IDs and the number of data points for that flight were found per runway. Then we found the mean and standard deviation of the number of data points for each runway. Using these values, we were able to calculate above which value of the number of data points a flight could be considered an anomaly. This was done by converting the number of data points to a Z-score and checking if it was above the Z-score for 2 standard deviations from the mean. It was chosen that 2 standard deviations were the desired accuracy since this would give a high likelihood of the detected anomalies being outliers whilst including a sufficient amount of flights. Finally, the anomalous flights and their number of data points were printed to a text file, such that they could be used for comparison later.

2. *Detection using runway approach*

The second approach considers the final approach to the runway on the Instrument Landing System (ILS) to detect anomalies. An example of six approaches on this ILS is shown in Figure 2. As can be seen, not every flight performs the same final approach. The most obvious variations occur in the horizontal direction perpendicular to the runway where large oscillations around the runway centerline (indicated by the dotted line in Figure 2) can be observed. As an example, consider flights AAB29B and AAB453 in Figure 2 in the straight-down view of the approach. They show very large oscillations around the runway centerline.

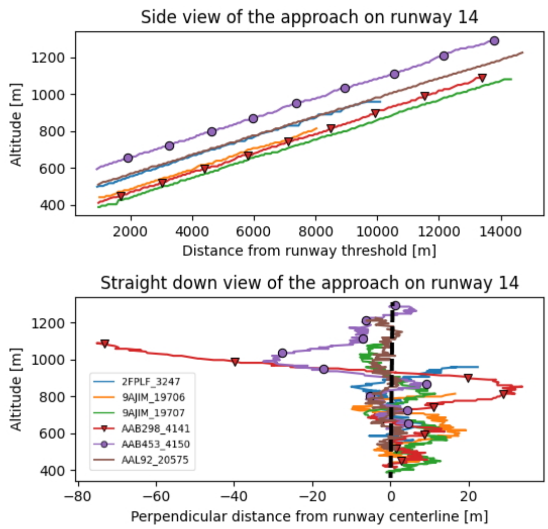

Figure 2: The distance and altitude on the final approach to runway 14 of six flights. The centerline of the runway is indicated by a dotted line

To capture the variations in these final approaches, the standard deviation of multiple parameters was computed over the entirety of the final approach. The parameters that were considered were the horizontal approach angle, the vertical deviation from the glideslope and the speed on the glideslope. The horizontal approach angle and vertical deviation were chosen because these parameters can be geometrically straightforwardly calculated from the data. The speed on the approach was chosen because it can give insight into the stability of the approach independently from the aircraft's path.

The standard deviations of the parameters were calculated for the values the flight would have if it followed a 'perfect' approach. A perfect approach is one with no distance from the centerline, a constant vertical approach angle (3.3° on runway 34 and 3° on runways 14 and 16 [10]) and a constant speed. This calculation was performed using Equation 1:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{1}$$

In Equation 1, x_i is the observed value of the parameter, μ is the value the parameter would have on a perfect approach and N is the number of data points.

Anomalous flights could then be detected by finding the flights with the highest standard deviations in their parameters compared to the other flights. For a flight to be considered an anomaly, it should be more than two times the standard deviation away from the mean. This two times standard deviation boundary was determined by inspecting the trajectories manually and determining the parameter such that visually anomalous flights were deemed to be anomalous by the statistical analysis.

C. Anomaly Detection using Computationally Expensive Methods

Using statistical analysis, as discussed, provides a simple and fast method to identify which flights have more data points and are hence more likely to be anomalous. Due to the high number of flights that are being analyzed, it is not possible to manually check each flight. To be confident in our results, the group decided that validating our statistical analysis is necessary.

1. *TensorFlow neural network*

Before the method used is described, it is important to discuss why this specific method was chosen over others. Aside from statistical analysis, analyzing the ADS-B data on a large scale can best be done with a form of machine learning. It was decided that using a convolutional neural network to perform supervised machine learning with a Python library the best option. In addition, by doing this even more information than just nominal or anomalous trajectories can be extracted from the data.

The main approach of this method was to render graphs of the trajectories of the flights, consistently, and save these images to disk such that they can later be used to train the neural network. It was important to be consistent with the graphs so that the trained neural network could accurately identify patterns of the trajectories for the data set later. Consistency can be achieved in multiple ways, the two most logical options include centering all of the flight patterns around a specific location in space, such as the start of the runway. Another option is to center each trajectory image around the center of the trajectory. The advantages of the first approach, centering the image around the start of the runway, including a more simple pre-processing stage. With this method, the size of the image can be easily set; however, with this method, one would need to create a new training set for each runway/airport that one wants to analyze because the landing direction and location of holding patterns will change depending on the runway. For that reason, despite being the technically harder procedure, the second approach has been taken.

With the end goal being that, all the images are the same size and scale, the group found the geometric center of the flight path for each image by taking the average of the minimum and maximum longitude and latitude respectively. Then, to keep all the images in the same aspect ratio, it was decided that from that middle point the image would show 0.75 degrees latitude and longitude in each direction. The number 0.75 was chosen such that the image is not too large, causing unnecessary workload for the computer, and such that even the geometrically large trajectories still fit on the image.

For training, the trajectories of 400 flights, along with their corresponding labels were passed into *TensorFlow* and this input data was used to build a neural network of several layers of nodes. To test the accuracy of the data set, a testing data set was compiled with about 111 different flights and the neural network was used to identify the closest match for the pattern. For each flight, the pattern from the neural network was compared with the correct label, and this was used to determine the accuracy of the neural network for that test data set.

The number, and type, of hidden layers are very important. The accuracy of the model is heavily dependent on this. As a starting point a tutorial on the TensorFlow website [11], in which an image recognition CNN, was used. From there the group experimented with different numbers of layers, as well as varying optimizers, loss functions and activation functions. The model architecture that the group decided on made use of three layers, all using the rectified linear (ReLU) activation function in combination with max pooling and dropout processes. The model was then compiled using the Adam algorithm with the Sparse Categorical Cross Entropy loss function. The exact implementation can be found in the public code [9].

The trained neural network model was then used to predict the patterns for all the flight trajectories in the data set and this table of results was saved as a .csv file and contained the headers 'Flight ID', 'Prediction' and 'Anomaly T/F', with the last column containing True/False dependent on whether a normal prediction was made for the corresponding trajectory.

2. *Density-Based Spatial Clustering of Applications with Noise*

The previously described approach requires a considerable amount of work and processing resources to accomplish. Therefore, an additional, simpler approach is also presented. The anomaly detection is done using the Density-based spatial clustering of Applications with Noise (DBSCAN) algorithm, a fast clustering algorithm, which creates clusters based on the density of points, in our case flight paths [12].

The algorithm is implemented in the Python *sklearn* library and thus provides a simpler alternative to the previously described neural network. To perform the algorithm, the DBSCAN function of the library needs to be invoked as described in the libraries' documentation and the parameters, namely the minimum number of samples per cluster and the epsilon, which dictates the maximum distance that trajectories can be apart from each other to be placed in the same cluster, adjusted to achieve acceptable results [12].

After the execution, the program returns data files, which include the flight IDs with their respective clusters, marked by numbers. The flights marked with *-1* are considered anomalous.

III. Results & Discussion

This section focuses on presenting the results and the discussion on them. Firstly, subsection A presents the results of the anomaly detection using the number of data points, followed by subsection B, which gives the results and discussion on anomaly detection using the runway approach. Furthermore, subsection C provides the results and discussion of the neural network followed by subsection D with the results of the DBSCAN clustering algorithm. Lastly, subsection E discusses the comparison between the neural network and the heuristic approaches.

A. Anomaly Identification based on the amount of Data Points

In this section, the detected outliers are presented. Runways 14, 28 and 34 are the runways that are the most used for landing in the time frame monitored. Runway 16 is used 9 times meaning we do not have enough data to draw a conclusion or find the outliers. Runways 10 and 32 have not been used for landing at all. For runway 14 we see in Figure 3 that the distribution resembles a normal distribution.

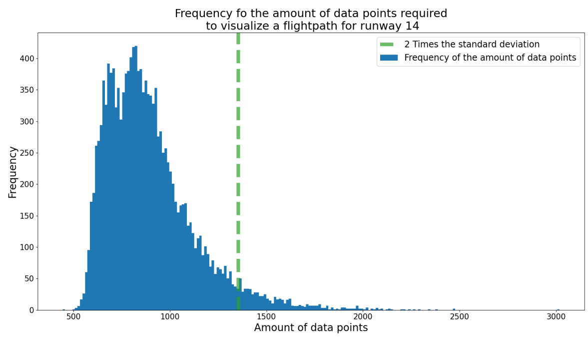

Figure 3: Frequency of the amount of data points for runway 14, μ =893, $\sigma = 230.156$

In Figure 3, the distribution is shown for which the standard deviation and mean can be found. For runway 28 and 34 we have a similar distribution and with different means and standard deviations. As shown in Table 1.

Table 1: The mean, standard deviation, number of flights and number of outliers per runway for the amount of data points.

Runways	**14**	**28**	**34**
Mean	893	1029	1070
Standard deviation	230.156	280.424	238.684
Number of flights	14 457	3166	1598
Number of outliers	667	153	59

In Figure 3 the green line marks the point that is 2 standard deviations from the mean. Anything to the right of this line is considered to be an anomaly. Using this method for runways 14, 28 and 34, the data shown in Table 1 was found.

The trade-off between speed and accuracy for this method shifts more towards the side with high quickness and lower accuracy. The method gives the first indication of possible outliers, which can be used for further analysis. The loss of accuracy also stems from the dependency on many factors. For example, an aircraft that flies faster has fewer data points, due to a larger distance between them.

B. Anomaly Identification based on the Final Approach

In this , the results of the anomaly identification using the final approach will be considered. First, the statistical distribution of all the approach parameters will be shown. Second, the outliers derived from those distributions will be discussed.

In Figure 4 through Figure 6 the distribution of the frequencies of the standard deviations of all the parameters can be seen. In Figure 4 the frequency of the standard deviation of the vertical approach altitude is shown. It can be seen that most flights have a small deviation, close to zero. However, this distribution has an extremely long tail with outliers stretching all the way to a standard deviation of 300 m. Figure 5 and Figure 6 show the frequencies of the deviations in speed and horizontal track angle. These are more normally distributed but both seem to contain small concentrations of data towards zero.

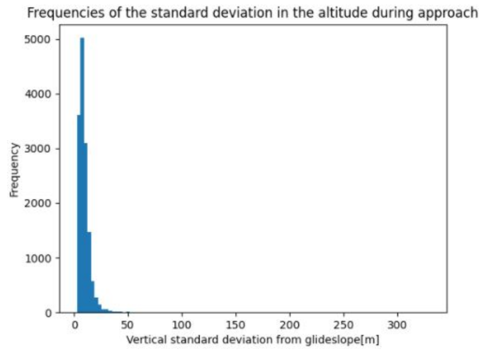

Figure 4: The frequencies of the standard deviations of the vertical deviation. $\mu = 9.89$, $\sigma = 12.38$

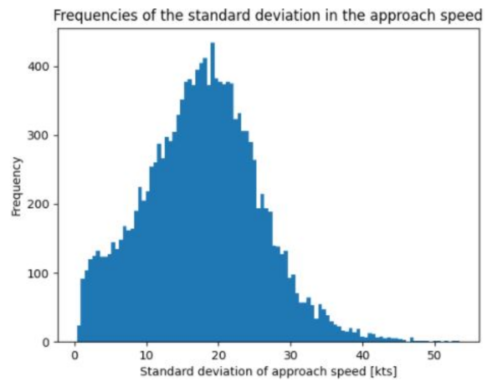

Figure 5: The frequencies of the standard deviations of the speed. $\mu = 17.68$, $\sigma = 8.0$

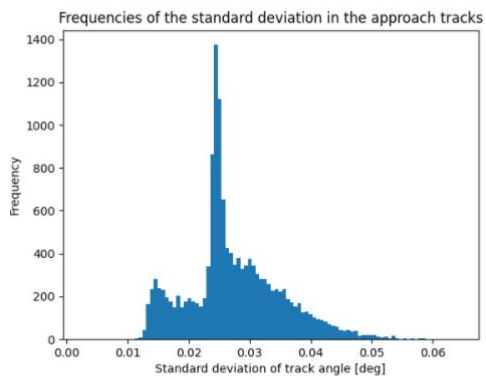

Figure 6: The frequencies of the standard deviations of the horizontal angle. $\mu = 0.027$, $\sigma = 0.008$

In Figure 7 through Figure 9 the approaches that were determined to be anomalous by the method in section II are plotted. No flight id's are presented because of the high number of flights but each color represents a different flight. There were 148, 409 and 557 outliers respectively for the outliers detected using the vertical approach, speed and horizontal angle parameters.

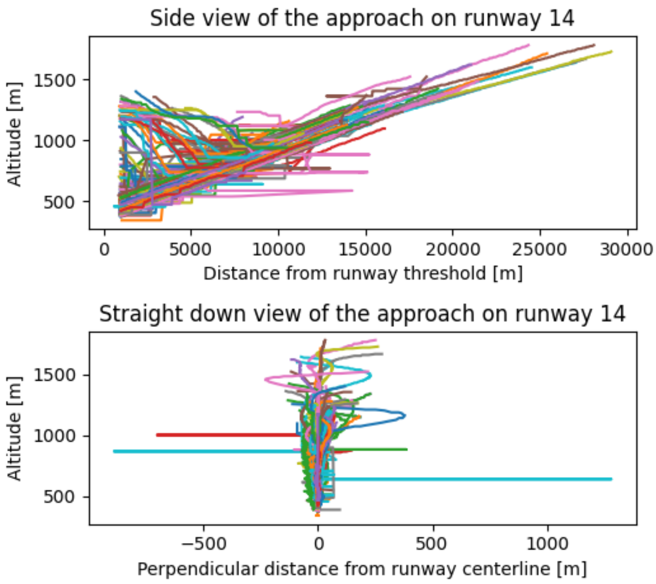

Figure 7: The anomalous approaches as detected by the vertical deviation.

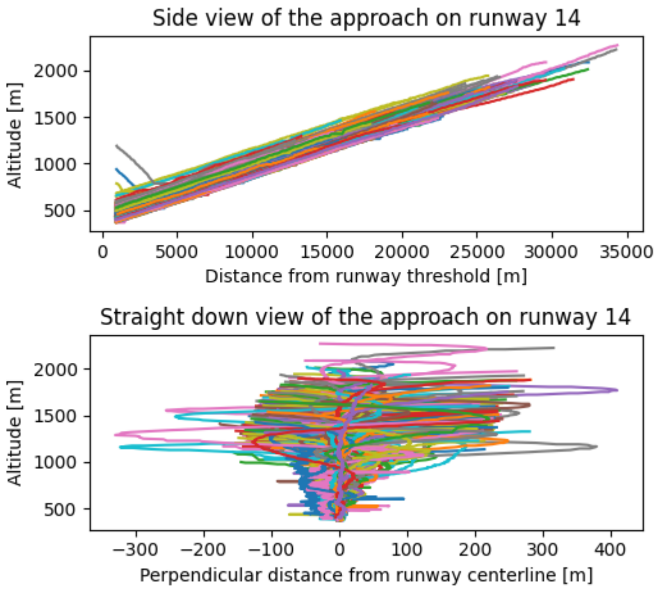

Figure 8: The anomalous approaches as detected by the variation in speed.

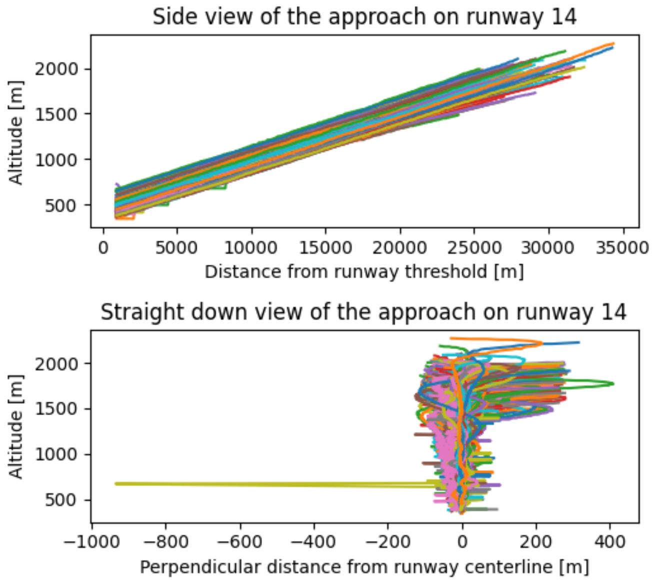

Figure 9: The anomalous approaches as detected by the variation in horizontal approach angle

The goals of the method were to detect anomalous flights using their stability on the final approach. This was performed by looking at three parameters: the vertical deviation, the speed variation and the deviation in horizontal track angle.

By considering the anomalies as detected by the vertical deviation, three kinds of anomalies were detected: first, flights containing noisy altitude data, second, flights containing go-arounds and third, flights with slightly differing average vertical track angles. These are all desirable flights to be selected as anomalies by the method although they are all anomalous in slightly differing ways.

The anomalies detected by their speed variation show a strong tendency to also be unstable in the horizontal direction, with deviations of up to 400 m. There is almost no correlation with vertical deviations, except for the flights initiating a go-around due to their unstable approach.

The anomalies detected by looking at the horizontal deviations are the weakest. Most of these anomalies could pass as a normal approach. This is further confirmed by the fact that these anomalies include almost no go-arounds. This means that these approaches were also judged to be stable by the pilots and therefore not extremely anomalous. The weakness of this method could be explained by the fact that some of the approaches wrongly include the final turn to the runway (these turns can be seen in Figure 9 in the top right of the straight-down view of the approach, there is a large concentration of flights in the 100 to 350 m range). This turn heavily influences the average horizontal angle as computed by the method and throws off the anomaly detection.

In general, the three methods of anomaly detection on the final approach seem generally independent from each other with only a few flights being unstable on all three parameters (mostly flights containing go-arounds, which confirms that those approaches were also deemed unstable by the pilots). The only small coupling that is present is between the variation in speed and the variation in horizontal track angle.

C. Convolutional Neural Network results

This section details the results that were obtained from using the Convolutional Neural Network to classify flight trajectories into distinct patterns. First, pre-processing of a small subset of the data set was done, as is detailed in the method.

The images resulting from the pre-processing stage are displayed in Figure 10. At least one example, of each of the categories the group decided a flight can fall into, has been shown. These images are the inputs of the neural network.

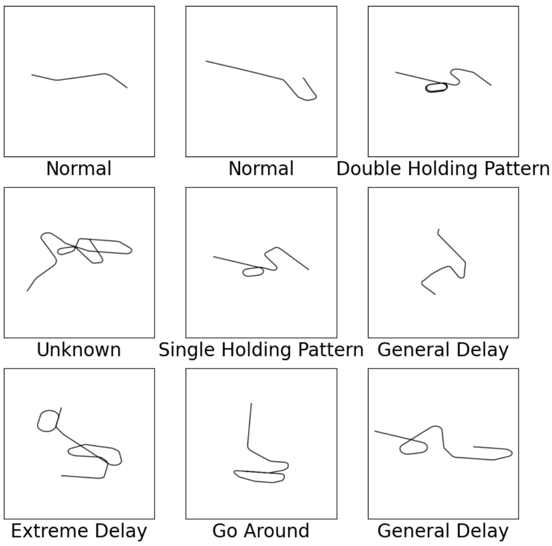

Figure 10: Images of the flight trajectories after pre-processing. Examples of all the categories the group decided a flight can fall into have been labeled.

Although the main analysis being completed is whether or not a flight is anomalous (anything other than Normal), within the set of anomalous flights different categories have been included to show the capabilities of the neural network.

The training aspect involved training the model over about ten epochs, or rounds. In each epoch, images from the training data set were used to train the different underlying layers of the neural network and images from the test data set were used to check whether each modification to the model yielded higher accuracies or not.

The accuracy after the final epoch was about 88%. This means that the neural network has a 88% chance of identifying the right pattern for a given trajectory. Once the model was trained, it was ready to be used to predict and separate anomalous trajectories in the full data set from normal trajectories.

After using the neural network to perform this separation, the group manually inspected all of the images to validate that the accuracy is what the neural network stated it to be. This was done by looking at both the false positives and the false negatives. As expected, since the neural network has an accuracy of approximately 88%, about 10% of the images classified as normal were anomalous. As well as this, the number of false negatives was inspected to be very small, in the order of 5%.

The use of a Convolutional Neural Network in identifying anomalies is a method that has been in use in analysis for a while. It is relatively complex to set up but not complex enough that it requires a lot of configuration before meaningful results can be derived.

One advantage of using a Convolutional Neural Network is that given sufficient training and testing data, neural networks can be trained to very high accuracy and used very reliably for anomaly detection. Another advantage is that as more data is obtained for training, the models will get more and more accurate and can adapt to changing trends in the data.

However, one disadvantage is that the collection of and processing of data used to train the models require highly available computing resources, most of which are quite expensive with the scale of data that needs to be processed. Another disadvantage is that as models become more accurate at anomaly predictions, the amount of time required to tweak and retrain the Convolutional Neural Network will also increase which could be undesirable when results are needed quickly.

D. Anomaly Detection using Density-Based Spatial Clustering of Application with Noise

This section elaborates on the results achieved from the DBSCAN clustering algorithm and its outliers. The clustering algorithm was performed multiple times with varying parameters until an acceptable balance between clusters and outliers was achieved. These parameters were determined to be:

- Minimum number of samples per cluster= 125
- ϵ= 1.2

With the parameters, the algorithm provided 12 clusters of flights with the distribution presented in Table 2. The algorithm detected 2981 outliers, which is approximately 18% of the total number of flights (16710 flights[a]). The results of the clustering are shown in Figure 11, with the color-coded clusters and the outliers presented in gray.

Table 2: Distribution of flights per cluster using the DBSCAN

Cluster	Number of flights
1	1328
2	427
3	2607
4	279
5	3791
6	234
7	311
8	2275
9	673
10	306
11	729
12	229
Outliers	2981

[a]Less than the total number of flights, due to additional filtering to filter out data points which provided errors in the algorithm

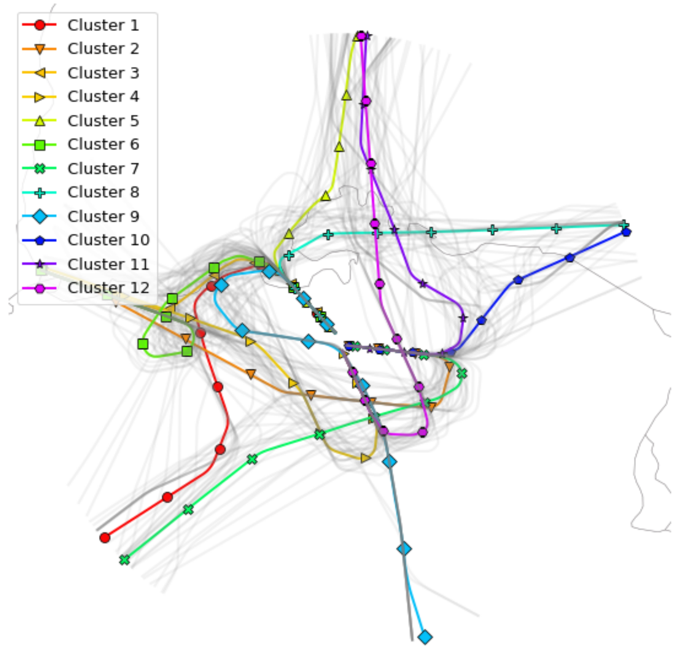

Figure 11: Results of the DBSCAN clustering of the data set with the minimum sample parameter = 125 and ϵ=1.2

The results provided by the DBSCAN clustering algorithm are seen as representative of a fast and simple outlier detection using machine learning. There are three main cases, seen in figure Figure 11, which provide the downsides of the algorithm:

- Flights that follow the clustered path are detected as outliers because they took a sharper or wider turn
- Dense areas of flights, which could be considered nominal, are detected as outliers due to the low number of flights
- Flights with holding patterns, which should be considered anomalous, are clustered as nominal due to their density

These downsides can be to some level adjusted for by changing the algorithm parameters, as was done in this case, but by changing the parameters one issue becomes less prevalent while the other increases (for example by decreasing the minimum number of samples, flights which can be considered nominal will become clustered, but anomalous flight clusters will become clustered as nominal as well), therefore a balance between the problems has been chosen.

Due to the downsides listed the results offer worse performance than the neural network. While it did detect a comparatively large number of anomalies, lots are flights with flight paths different than the majority, but should not be classified as anomalous, due to the lack of such behavior (eg. lack of holding patterns or go-arounds). Corrections for these errors could be made but with it the processing requirements would increase, defeating the purpose of a low processing power machine learning algorithm.

E. Comparison of the Heuristic Approach with the Neural Network

To see if heuristic approaches which require less computing power are suitable to be used as anomaly detection, the results of the approaches will be compared to the output of the neural network to find the overlap they provide. Each of the methods outputs a file containing the anomalous flight IDs. These files were cross-checked for inclusion in the neural network output. The overlap results are found in the following table.

Table 3: Overlap of the anomalies detected by alternate approaches compared to the neural network

	Data points	ILS	DBSCAN
Overlap	45.28%	4.31%	25.19%

The second point of comparison is the number of detected anomalies, presented in Table 4, this shows that the methods varied a lot in the number of anomalies detected. Under the assumption that the neural network is accurate, it can be concluded that the statistical methods used are not very useful for identifying anomalous flights.

Table 4: Number of anomalies detected by each method

Neural network	Data points	ILS	DBSCAN
1941	879	1114	2981

IV. Conclusion & Recommendations

This section will explore the feasibility of the alternate anomaly detection approaches compared to an accurate baseline, the neural network. The neural network anomaly detection has been selected as the baseline for anomaly detection as it has a high precision, which is the result of its high processing power. As shown in the result section.

From Tables 3 and 4 in the results section, it can be discerned, that the most accurate alternative detection method is detection using the number of data points with an almost 50% overlap. While this number is large looking at the percentages it fails when it is taken into account that this is only around 240 of the 1941 anomalies detected by the neural network. The DBSCAN, which came in second on the overlap, features about 750 of the same anomalies as in the neural network which seems promising, however, it detected almost 2250 false anomalies. In conclusion, none of the heuristic approaches covered in the paper, including a low computational power machine learning algorithm, can stand up to a task-specific neural network and is thus recommended to not use them for accurate flight path anomaly detection.

Our recommendations would be, to continue the research with more heuristic methods that may provide better results, such as velocity or weather, or to turn the research direction into developing a computationally simple neural network. In addition to this, the accuracy of the neural network can also be improved upon.

References

[1] Sun, J., *The 1090 Megahertz Riddle. A Guide to Decoding Mode S and ADS-B Signals*, TU Delft, 2020.

[2] Schäfer, M., Strohmeier, M., Wilhelm, M., Martinovic, I., and Lenders, V., "Bringing up OpenSky: A Large scale ADS-B Sensor Network for Research," Tech. rep., TU Kaiserlautern, University of Oxford, 2014.

[3] ICAO, "Forecasts of Scheduled Passenger and Freight Traffic," , 2021. https://www.icao.int/sustainability/pages/eap-fp-forecast-scheduled-passenger-traffic.aspx.

[4] Lishuai, L., and Hansman, J., "Anomaly detection in airline routine operations using flight data recorder data," *MIT ICAT*, 2013.

[5] Olive, X., Basora, L., Viry, B., and Alligier, R., "Deep Trajectory Clustering with Autoencoders," *Proceedings of the International Conference on Research in Air Transportation*, 2020, pp. 1–8.

[6] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Association for the Advancement of Artificial Intelligence*, Vol. KDD-96: Proceedings, 1996, pp. 226–231.

[7] Olive, X., "Traffic – Air traffic data processing in Python," , 2018. https://traffic-viz.github.io/index.html.

[8] Olive, X., "Traffic, a toolbox for processing and analysing air traffic data," *Journal of Open Source Software*, Vol. 4, 2019, p. 1518. doi:10.21105/joss.01518.

[9] Group D05, "Aircraft-anomaly-detection," , 2021. https://github.com/TUD-Group-D05/aircraft-anomaly-detection.

[10] vACC Switzerland, "Airports and Charts - LSZH - Zurich (major airport)," , 2021. https://www.vacc.ch/en/airports_and_charts.

[11] TensorFlow, "Convolutional Neural Network (CNN)," , 2021. https://www.tensorflow.org/tutorials/images/cnn.

[12] Yuan, G., Sun, P., Zhao, J., Li, D., and Wang, C., "A review of moving object trajectory clustering algorithms," *Artificial Intelligence Review*, Vol. 47, No. 1, 2017, pp. 123–144.