# Shape Analysis of Flight Trajectories Using Neural Networks

Colton Gingrass* Dashi I. Singham† Michael P. Atkinson‡
Naval Postgraduate School
Operations Research Department
1411 Cunningham Road
Monterey, CA 93943, USA

October 21, 2020

**Abstract**

The recent widespread implementation of ADS-B (Automatic Dependent Surveillance - Broadcasting) systems on aircraft allows for improved monitoring and air traffic control management. As part of this monitoring, it is important to be able to detect unusual flight trajectories due to weather events, detection avoidance, aircraft malfunction, or other activities which may signal anomalous behavior. Given the large volume of ADS-B data available from aircraft around the world, the ability to automatically determine the shape of the trajectory and identify anomalous behavior is important to reduce the need for human identification and labeling. We develop a neural network model for multi-category classification of the shape of the trajectory using features derived from a large ADS-B data set such as bearing and curvature. The results suggest promise in differentiating common trajectory shapes using key factors, with the accuracy of the classifier being comparable to human accuracy.

*Keywords*: trajectory analysis, anomaly detection, air traffic flow, supervised learning.

## 1 Introduction

Flight monitoring for anomaly detection is an active area of research. Given the recent widespread deployment of ADS-B technology (Automatic Dependent Surveillance-Broadcasting),

*ENS, US Navy, Operations Research Department, 1411 Cunningham Road, Monterey CA, 93943
†Corresponding author, dsingham@nps.edu. Research Associate Professor, Operations Research Department, 1411 Cunningham Road, Monterey CA 93943.
‡Associate Professor, Operations Research Department, 1411 Cunningham Road, Monterey CA 93943

there is a large amount of open-source data available for analyzing aircraft trajectories. This fairly new technology relays information on the aircraft's position, altitude, speed, and other attributes from equipped aircraft in real time with precision. Given that in 2020 the Federal Aviation Administration has standardized and enforced the use of ADS-B on all aircraft in controlled airspace, the large amount of data produced and recorded by this system can be used to establish baseline criteria for flights that deviate from normal air traffic control patterns. Automated methods of classifying and evaluating trajectories can serve to limit the amount of human effort required to study flight behavior, and this has resulted in much research to automatically detect anomalies in flight paths. There are many types of anomalies of interest, for example, aircraft malfunction during takeoff and landing, detours taken to avoid weather events, avoidance of potential collisions for air traffic control, or flights attempting to avoid detection or deviate from their planned trajectory. The ability to use machine learning to automatically flag potentially anomalous flights would allow analysts to narrow their search and quickly diagnose potential problems.

Our focus is on detecting and classifying unusual flight trajectory shapes. Most flights take a generally straight path from the origin to the destination. However, a small percentage of flights may travel in unusual patterns, such as multiple loops, figure-eights, or taking a jagged path towards the destination. Shape classification is important because the shape may be a proxy to detect an underlying activity of interest; for example a weather event could significantly affect flight trajectories, or the shape may reveal sightseeing or flight training activities. Furthermore, the ability to quickly identify unusual flight trajectories may aid in national security efforts to investigate anomalous behavior across the globe, which often rely on large efforts by human analysts to monitor flight data. Detection avoidance, smuggling, or a military maneuvering exercises are all activities of interest that could affect the shape of a flight trajectory. Our goal is to identify this small percentage of potentially anomalous paths and automatically differentiate them according to their trajectory shape. We employ a neural network to help account for the underlying complexity in translating from ADS-B

information to a shape classification.

In conducting shape analysis, it is natural to look at the sequence of directional changes made by the moving object. For example, an object traveling in a straight path will have small or zero changes in heading between different observation points. In other trajectories, a sharp turn will register as a single large directional change. While studying the sequence of angles has been used to model shape trajectories in video surveillance or studies of vehicular motion, it appears natural to apply this idea to ADS-B data because these changes in direction can be easily calculated. Studying this "change in heading" isolates the focus on the shape and has the effect of normalizing the observations across different locations, and also normalizing across the general direction a trajectory could take.

In addition to including the changes in direction as inputs to the neural network, we use estimated values of the curvature at the observation points in the data. Curvature is a measure of the deviation from a straight line, and can be estimated at each observation using the points immediately before and after the current point. It takes into account change in direction relative to distance traveled, and so provides different information than the change in heading. Both the change in heading and the curvature can be easily estimated at each point using the latitude and longitude coordinates of ADS-B data.

Additionally, we consider many other input variables based on the location data, normalized by total track distance, to focus on the shape and not length of the flight. For example, we can include normalized distance between the starting and ending point of the trajectory to distinguish between straight line tracks and loops. We also compare the median location with the halfway location, where the median location is the median latitude and median longitude across all points on the track and the halfway location is the point on the track where the aircraft has traversed 50% of the track distance. Note the halfway location must lie on the actual track, whereas the median location need not. For example, a circular track will have a median location in the center of the circle while the halfway point will lie on the circle itself opposite the starting location. Four key locations of interest are the starting

point, ending point, halfway point, and the median location. We calculate distributional properties of the normalized distances between these points and all other points on the track, and the distributional properties of the headings between these points and all other points on the track. Such information can be readily calculated and can help isolate different types of shapes.

To the best of our knowledge, our work is the first to apply neural networks to the shape analysis problem in aircraft trajectory using ADS-B data. There are numerous streams of literature to conduct shape analysis of trajectory data as discussed in Section 2. Many of them involve collecting features of each trajectory, and then using clustering to group together and classify trajectories with similar shapes in an unsupervised setting, while our focus is on developing an extensive labeled data set as input to a neural network. Section 3 describes the data we use and the types of trajectory shapes we want to classify. We classify thousands of tracks falling within nine obvious shape categories (called *standard* tracks) so that the labeled data can be used to train the neural network model. We also consider data that does not fall into one of the standard shapes (*hybrid* tracks), as there can be high variability in flight trajectories. Section 3 also describes the process for choosing input factors and tuning the hyperparameters of the neural network model. We employ a Nearly-Orthogonal Latin Hypercube (NOLH) sampling method for testing various hyperparameters of the neural network to ensure we cover a space of possible options.

Section 4 presents the results of the model. The model performs favorably on the standard data set overall with a weighted F1-score of 88%, and an unweighted average across shape categories of 62%. Given that we observe only 80% accuracy in human labeling due to variability in trajectory appearance, the neural network appears promising in filtering out potentially anomalous behavior. The model is able to clearly differentiate straight line tracks which are a significant part of the data set. It becomes harder to differentiated the unusual classes, but we note that the model performs much better than a random chance assignment, and misclassifications often occur within similar shape types yielding credence to our choice

of inputs. Because much of the ADS-B data falls outside the nine main shapes, we conduct a separate analysis on the hybrid shapes to assess the performance of the trained model on these unclassified trajectories. We find that the model does a relatively good job of separating tracks that generally travel in an outward direction from start to end (one-way trips) from those that return to the starting area (return trips). In some cases, the model is able to further differentiate particularly unusual shapes, and we demonstrate using some real examples. Section 5 concludes and suggests avenues for future work.

# 2 Literature Review

We summarize literature related to three main areas. First we discuss papers which use machine learning and neural networks to analyze trajectory data. Next, we focus on research performed on aircraft data for determining anomalous behavior for air traffic control purposes. Third, we describe work performed to analyze the shape of trajectory data, which is of particular interest to us.

There is much work that uses machine learning methods to analyze various types of trajectories. In addition to flight trajectories, vehicle or pedestrian movement is a major area of interest. [1] develop a semi-supervised learning method for classifying pedestrian movement. A human is used to classify a small number of paths as normal, and spline methods are used to model the paths. A single-class is used to model normal paths, and hence anomalous paths are detected by filtering out observations that fall far from a Gaussian mixture model. [2] look for anomalous trajectories taken by people walking through a scene in the context of video surveillance. Common pathways are modeled as nodes of clusters, and the first check is whether a person is walking on a path. The next check is whether the person is moving at a normal velocity, and the last check is for curvature to see if the object is moving in an unusual way. Paths are classified as anomalous if they fail one of these three checks. We will also use track curvature as one of our input variables to detect interesting shapes. [3]

provide a survey of surveillance modeling when vision or video data is available. They detail numerous methods for analyzing tracks and finding anomalous behavior. Additionally, they describe methods for online classification and other variations of this problem.

[4] provide an extensive survey of two types of mobility prediction problems: future location prediction (predicting where an object will be at some future time) and the trajectory prediction problem (what is the anticipated movement in the future). [5] uses a recurrent neural network to train sequence-to-sequence models for predicting where a vessel will arrive based on studying past trajectory data over a space divided into a grid. The model employs AIS (automatic identification system) data and returns the most probable arrival location, in addition to alternative arrival locations. Clustering methods are also a major tool used in motion mapping and prediction, see for example, [6]. Trajectory data can be mapped to a feature space, sometimes by using frequency domain methods, and clusters formed to group together similar types of motion. [7] use an autoencoder to derive features of motion and use clustering accuracy to separate motion patterns across people moving in different locations.

There has been substantial research on detecting anomalous flight behavior. Many specific techniques have been designed to analyze air data for preventative safety measures in commercial aircraft. [8] employ a novelty detection method to search for abnormalities in the descent of commercial aircraft. A Gaussian mixture model and $K$-means classifier are used to determine a one-class method. $F$-score tests can be used to help with feature selection, but here are used to look for differences between the normal and abnormal data sets. Similar to the general motion mapping methods, there has also been a stream of research which uses clustering to group flight trajectories and identify outliers that may be anomalous, for example, in determining problems during takeoff and landing. Safety events can also be identified using large-scale data sets [9], where both fleet level and flight level anomalies were detected and validated by experts. The method in [10] employs a data transformation stage which consists of taking all information associated with a particular track and converting it to a single vector that can be input to the clustering method. Next, principal component analysis

is employed to reduce the dimension of the data, and then a clustering algorithm is used to classify observations and spot anomalies. Related work is described in [11, 12, 13, 14]. Real-time anomaly detection using a logarithmic multivariate Gaussian model is described in [15].

Anomalous behavior in air traffic management is also of high interest. The major method used is clustering to group together similar types of flight patterns. [16] employ a data driven method to assess the health of an air traffic control space using trajectory clustering to determine if incoming flights are following assigned flight patterns. They identify the turning points in a trajectory, model trajectories as sequences of waypoints, and cluster across the turning points and sequences of waypoints to group trajectories. [17] look at flight traffic for air traffic control management for ADS-B data in a specific region in France. The authors first use a clustering algorithm to classify all the traffic flow, and then use an autoencoder to try to determine anomalous behavior. Furthermore, the authors attempt to distinguish between weather and deconfliction as causes of the anomalous behavior. [18] employ clustering methods to find aircraft trajectories which may or may not deviate from the scheduled flight plan. They then use a hybrid of two probabilistic methods to offer improved predictions for aircraft. [19] employ random forests to predict whether aircraft rerouting requests will be operationally acceptable by air traffic control officials, while [20] look at new energy metrics using an unsupervised method to identify anomalous behavior in general aviation flights which have higher safety risks than commercial flights. [21] develops an incremental learning algorithm to perform daily updates to an anomaly detection model which identifies unusual arrivals of aircraft to airports. The model works by defining breakpoints to model a centroid of a cluster of flights for common approach patterns and identifying flights that deviate from the bounds determined by the model across numerous dimensions.

Neural networks have also been used to predict various aspects of flight trajectories. [22] use neural networks to predict the vertical trajectory of aircraft, which can be harder to predict than horizontal movement. They use a small neural network fit using 142 trajectories

and a single hidden layer. [23] create a simple neural network (one hidden layer with 5-10 neurons) from data to build predictive models for aggregate air traffic flow at an airport. [24] develops a neural network to attempt to automate air traffic control decisions. [25] constructs a neural network to determine the best sequence of arrivals for incoming aircraft with different priorities when fast decisions are needed. [26] models and predicts air traffic flow using a combined model with a convolutional neural network to model the spatial components, and a recurrent neural network to model temporal dependence, while [27] use a Long Short-Term Memory network to predict aircraft trajectories using ADS-B data.

Finally, we look at research on shape analysis of trajectories. [28] study video analysis of trajectories. The authors extract common features from the trajectories and use a clustering algorithm to group together similar types of motion. Anomalous behavior is found by looking for trajectories in a sparse space, or identifying those that behave differently from others in the same cluster. [29] differentiate between different types of objects on roadways by analyzing features of transformed trajectories, or motion pattern attributes. In air traffic control, [30] employ clustering to classify flights according to arrival patterns while taking into account the complexity of having multiple routes intersecting.

[31] was one of the first papers to study the identification of unique trajectory shapes independent of geographical area, and develops a new architecture for modeling and predicting non-linear motion patterns, in particular circular, polynomial, or sinusoidal patterns. This work relies on the fact that many motion patterns can be modeled using recursive functions. [32] uses the tangential angles at observation points and models them as a von Mises distribution. The von Mises distribution is a circular distribution and so is a natural choice for modeling directional angles. Building on past work in this area, [33] employs the speed of the object and uses kernel density estimation rather than fitting a traditional parametric distribution. They perform unsupervised learning via clustering to detect anomalous trajectories, while employing clustering and information theory.

[34] and [35] also model trajectories as a sequence of angles for shape analysis. They

employ circular statistics using a mixture of von Mises distributions in order to model the fact that not all angles will have the same properties, and develop an expectation maximization algorithm to estimate the mixture parameters. Then, each sample is assigned to the most similar distribution from the mixture and the sequences of assignments are run through a clustering algorithm to group and detect abnormal trajectories. This method was applied to video surveillance data to determine common trajectories of human motion. [36] applies a similar method to aircraft trajectories with the purpose of differentiating between manned and unmanned aircraft.

# 3 Neural Network Model Using ADS-B Data

To the best of our knowledge, [37] is the first to use supervised learning with neural networks for shape classification using ADS-B data, and this paper formalizes and extends that work. This section describes the data set we used to train the model, the input variables collected, and the process used to train the hyperparameters of the neural network model.

## 3.1 Data set

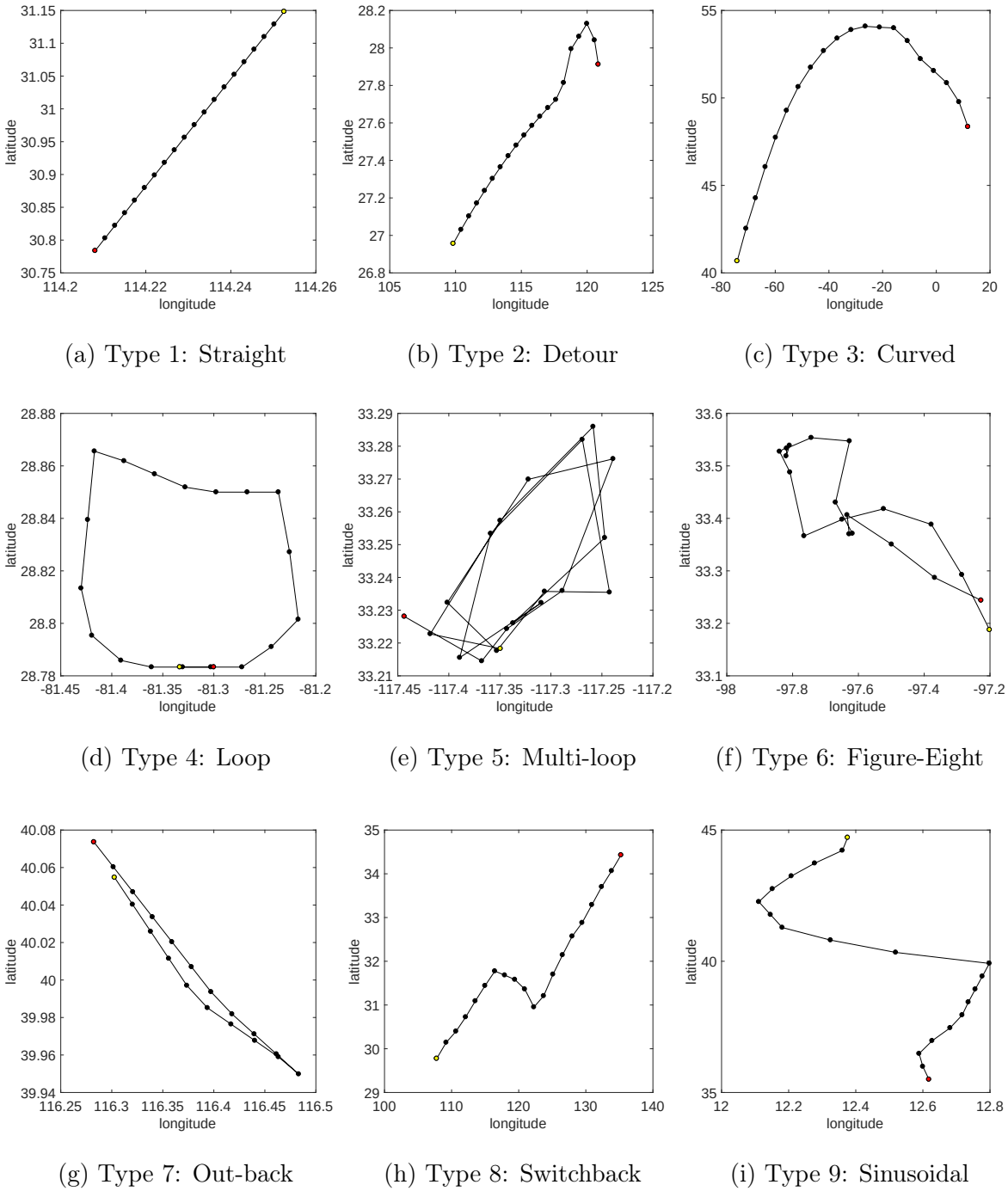
We relied on an ADS-B data set from October 2016 consisting of flights from all over the world. ADS-B data can be obtained from a variety of sources, including [38] and [39], and flights can be extracted from raw data using clustering methods such as [40]. We postprocessed the ADS-B tracks to filter in those which had well-defined starting and ending points, at least 10 location observations, travelled at least 5 miles, and had no time gaps exceeding 20 minutes.

In order to develop a comprehensive data set to train the network, we manually classified a total of 17,416 flight trajectory shapes. Initially, we went through a total of around 1,000 trajectories to form a basis for what the major shape types would be. We identified nine major shape types (called *standard* classes), and had a tenth label for trajectories that did

not fit into the nine major types (called a *hybrid* class). Most trajectories (around 72%) did not fit into an obvious shape and were placed in this hybrid class, and will be studied separately from the initial classifier on the tracks that clearly fit the nine standard types (28% of the total). There were 4,880 tracks classified as standard, and this data set formed the basis for training the neural network. There could have been many more shape types assigned, but we chose to focus on nine that could be clearly differentiated and appeared enough to be identified. This leads to a better training process by focusing on a clean data set where the input data meets strict criteria for being included in a standard class.

Examples of these nine key types are displayed in Figure 1, and we summarize them next. Each figure in this paper displays a modified version of the original track consisting of 20 points equally spaced in distance along the original track. We discuss this 20-point track approximation further in Section 3.2. Type 1 is a straight track without any major detours between the start and end points. Track 2 allows for a detour involving one turn, but otherwise is straight. Type 3 is a curved trajectory, or parabolic in extreme cases. Type 4 is a single loop, while Type 5 consists of multiple loops. Type 6 is a figure-eight pattern, while Type 7 is a flight that goes out to a location and then returns directly back to the origin. Type 8 is a single switchback pattern which involves two major turns in the middle of the trajectory, and Type 9 is a sinusoidal pattern that includes multiple changes of direction or multiple switchbacks between the start and end points. We will call Types 1,2 and 3 *one-way* types since they are the most common ways flights traverse from a starting to ending point, while Types 4 through 9 are *anomalous* types which may signal unusual behavior.

Once we determined these nine classes of trajectories, we had two of the authors who were new to the data classify the same initial data set (around 600 tracks) to establish a baseline for human accuracy. We found that 21.5% of the trajectories were given different classifications due to varying human interpretations of which category the track belonged to. Given the high variability in the tracks, many appeared to fall somewhere between two of the types and so human interpretation played a role into which shape was assigned, or whether




(a) Type 1: Straight (b) Type 2: Detour (c) Type 3: Curved

(d) Type 4: Loop (e) Type 5: Multi-loop (f) Type 6: Figure-Eight

(g) Type 7: Out-back (h) Type 8: Switchback (i) Type 9: Sinusoidal

Figure 1: Major shape categories of tracks.

it was deemed *hybrid*. For example, a mostly straight track with slight curvature could be classified as Type 1 (straight) or Type 3 (curved) depending on the observer. Similarly, a complex track which returns to its starting point could fall between a Type 4 (loop), Type 6

(figure-eight) or Type 7 (out and back). With additional classification experience and strict guidelines, it would be possible to reduce the human error rate over time. However, due to the variability between observers, in this paper we employed a dataset classified by a single person with significant labeling practice to reduce the effect of human variability which might lead to conflicting inputs to the model. We acknowledge that there may be other sources of human error even from a single expert labeler.

We further subdivide the hybrid class into those that were somewhere between Types 1, 2, and 3 (standard one-way trips), and those that were truly anomalous. To avoid training the classifier on ambiguous shapes, we call these hybrid trajectories that were similar to Types 1, 2, and 3 as *hybrid-1way* or H1, since they represent common one-way trajectories and 60% of the original 17,416 trajectories fell into this category. Essentially, there were a wide variety of tracks that generally appeared as one-way tracks but could not be obviously labeled as a standard shape due to minor detours, areas of non-standard curvature, or minor changes of direction at takeoff and landing. The remaining hybrid tracks that were not close to Types 1, 2, or 3 were classified as *hybrid-anomalous*, or HA.

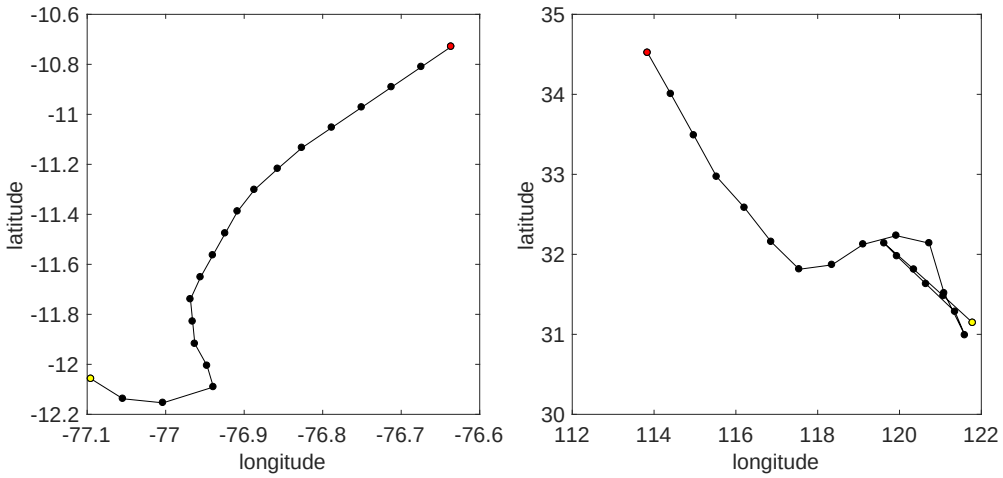



Figure 2: Examples of hybrid-1way (left) and hybrid-anomalous trajectories (right).

Figure 2 gives examples of trajectories classified as hybrid. The left plot shows one that

is hybrid-1way, in that there is both a detour and curvature which means it falls between Types 2 and 3. The right plot shows a trajectory that is hybrid-anomalous, because the unusual behavior in the bottom right of the figure means it clearly does not fall into any of the shape categories. The corresponding total track breakdown according to percentage of observations for the entire data set is in Table 1.

Table 1: Classification types and percentage of the overall data set. The hybrid-1way (H1) trajectories were similar to Types 1, 2, or 3 but could not be obviously classified as one standard shape.

| Standard | Hybrid |
|---|---|
| Type 1 (straight): 19.72%<br>Type 2 (detour): 2.15%<br>Type 3 (curve): 2.88% | hybrid-1way (H1): 60.42% |
| Type 4 (loop): 0.25%<br>Type 5 (multi-loop): 0.09%<br>Type 6 (figure-eight): 0.44%<br>Type 7 (out and back): 0.26%<br>Type 8 (switchback): 0.91%<br>Type 9 (sinusoidal): 1.32% | hybrid-anomalous (HA): 11.56% |

We use the standard data set to train the neural network, and split it into a 70% training set, a 15% validation set for tuning hyperparameters of the network, and display the results of the optimally trained model on a 15% test set which remains unused until the after the model has been finalized.

## 3.2 Input Variables

This section describes the input variables we derived from calculated features of a trajectory to identify the shape. Most of our input variables relate to heading and curvature, which are key features often used to identify trajectory shape (see Section 2). The raw data for each ADS-B track is a sequence of latitude and longitude points. From these points we derive 200 input variables associated with each track. The first two variables are the number of points on the track and total distance traveled. Large values of total distance usually correspond to one-way trips (e.g., Types 1, 2, 3). The remaining variables relate to distances, heading,

and curvature along the track. We compute distance between points and heading of the aircraft using standard great circle calculations [41]. An aircraft traveling from $(lat_1, lon_1)$ to $(lat_2, lon_2)$ covers a distance in miles of

$$dist = 3958.8 \arccos\left(\sin(lat_1)\sin(lat_2) + \cos(lat_1)\cos(lat_2)\cos(lon_2 - lon_1)\right).$$

The initial heading between two points $(lat_1, lon_1)$ and $(lat_2, lon_2)$ is

$$\begin{aligned} heading = \mod \Big( & \arctan 2\big(\sin(lon_2 - lon_1)\cos(lat2) \\ & , \cos(lat1)\sin(lat2) - (\sin(lat1)\cos(lat2)\cos(lon_2 - lon_1))\big) + 360, 360\Big). \end{aligned}$$

Our curvature calculation requires three points on a track. We use the Menger variant, which defines curvature as the inverse of the radius of the circle that passes through three points [42]. The three points also define a triangle with side lengths $a$, $b$, and $c$ and area $Area$. The circle radius is $\frac{abc}{4Area}$. We normalize this radius by total distance traveled on the track to give the final value of curvature used in our analysis:

$$curvature = \frac{4Area}{abc} \times \text{track distance}.$$

To compute curvature we use Euclidean distance for triangle sides $a$, $b$, $c$ rather than great circle distance because the curvature calculation assumes a Euclidean framework.

Curvature and heading information (specifically how the heading changes) provides valuable information for determining the shape. The heading changes and curvature should be very small for a straight line track (Type 1). A loop track (Type 4) should have roughly constant heading changes and curvature. For an out-back track (Type 7), the heading should remain fairly constant until an abrupt change when the aircraft turns around. The curvature on a figure-eight track (Type 6) should oscillate between large values near the top and bottom of the 8 when the aircraft turns around and smaller values in the middle.

Our first set of variables all relate to a modified version of the original track. The modified track consists of 20 points equally spaced in distance along the original track. We examine these modified tracks to enforce consistency as not all tracks have the same number of points and the points are not necessarily equally spaced. Recall all figures in the paper display this 20-point track approximation. From this modified track, we compute the heading between successive points and take the difference between headings at the 18 interior points as variables in our dataset. Additionally, we compute 18 curvature values at these interior points of the track and add them to the dataset. The dataset also includes summary statistics of the heading differences, absolute heading differences, and curvature. These summary statistics consists of the mean, median, various percentiles, standard deviation, skewness and kurtosis. Similar to how the von Mises distribution can be used to evaluate the distribution of a sequence of angles, we observe the distribution of our key input variables which are the change in heading and curvature. Finally, one variable specifies the index (0 through 19) with the largest absolute heading difference, in an effort to separate shapes with major changes in direction at different points in the trajectory. In total, the modified tracks generate 67 variables.

In addition to looking at the change in heading and curvature to determine the shape, we consider numerous other relationships between key locations on the track that might correlate with shape. For the remainder of this section any distance variable is normalized relative to the total distance traveled to focus on the shape of the track. These variables also correspond to the actual track, not the 20-point approximation version described above. Many of the remaining variables relate to four locations described in Section 1:

- *START*: the starting location of the track.
- *END*: the ending location of the track.
- *HALF*: the location on the track where the aircraft has traveled 50% of the total track distance.
- *MED*: median latitude and longitude across all track points. *MED* is the only location

that does not necessarily lie on the track.

We define six variables corresponding to the distance between all pairs in {*START*,*END*,*HALF*, *MED*}. The distance between *START* and *END* is usually large for one-way tracks (Type 1, 2, 3) and small for return tracks (Type 4, 5, 6, 7). The headings between pairs produces another six variables.

For each point on the track, we compute the distance to *START*. We then compute summary statistics for these distances and include them in our dataset. We repeat this for *END*, *HALF*, *MED* to generate 28 total variables. For a straight track (Type 1) the distances corresponding to *START* follow roughly a uniform distribution on [0,1] (recall we normalize distances relative to total distance traveled). For a loop track (Type 4), the distances corresponding to *MED* (the center of the loop) would be nearly the same for all points (approximately $\frac{1}{2\pi}$).

We generate similar variables using heading information rather than distance. For each point on the track, we compute the heading from *START* to that point. We then take the difference between successive headings and calculate summary statistics. We compute similar values for *END*, *HALF*, *MED* and repeat the process for absolute heading difference to generate 84 more variables. For a straight track (Type 1), the heading differences corresponding to *START* would be close to 0 for all points.

The final seven variables relate to curvature across non-successive points. First we compute the curvature between the following three points: the point located at 10% along the total track distance, the point located at 15% along the total track distance, and the point located at 20% along the total track distance. We repeat this for the following six percentile combinations: 45/50/55, 80/85/90, 10/30/50, 50/70/90, 25/50/75, 10/50/90.

Table 2 summarizes the variables in our dataset by category. The first column specifies whether we derive the variables from the original track or the 20-point approximation. The second column lists the metric: distance, curvature, heading. The third column provides a brief summary of the relevant variables, and the final column specifies the number of variables

in the category.

Table 2: Summary of Variables

| Track Type | Metric | Description | Number of Variables |
|---|---|---|---|
| Original | Number of points on track | – | 1 |
| Original | Total distance traveled along track | – | 1 |
| Original | Distance between {$START$,$END$,$HALF$, $MED$ } | values | 6 |
| Original | Heading between {$START$,$END$,$HALF$, $MED$ } | values | 6 |
| Original | Distance from {$START$,$END$,$HALF$, $MED$ } | summary statistics | 28 |
| Original | Heading Differences from {$START$,$END$,$HALF$, $MED$ } | summary statistics | 44 |
| Original | Absolute Heading Differences from {$START$,$END$,$HALF$, $MED$ } | summary statistics | 40 |
| Original | Curvature at percentiles on track | values | 7 |
| Approximation | Heading Differences | values; summary statistics | 29 |
| Approximation | Absolute Heading Differences | summary statistics; max index | 11 |
| Approximation | Curvature | values; summary statistics | 27 |

We originally tried many more input variables (around 1,000), and were able to obtain similar performance using the 200 presented in Table 2 and so chose the smaller model to avoid overfitting.

## 3.3 Model Details and Hyperparameter Tuning

Given the complexity of the data set and the potential relationship between the input variables and the output shape classifications, we employ a deep sequential neural network to perform supervised learning. In order to tune the network according to relevant hyperparameters, we use a design of experiments method to explore the space of possible networks and choose one with high predictive value by employing a sequence of Nearly Orthogonal Latin Hypercube (NOLH) experimental designs [43]. NOLH designs are space filling in that they allow for a multidimensional space to be efficiently searched at the expense of pure othogonality which is obtained in the usual Latin Hypercube designs [44]. This means that we do not need to try every combination of possible hyperparameters, but can ensure that we sample from different regions of the mulitdimensional space.

We need to choose the two major parameters of the network structure: the number of hidden layers (network depth) and the width of each layer (number of neurons). Additionally, we need to choose parameters of the optimization when fitting the network, including the number of epochs to run the optimization, the batch size for the gradient descent, and the learning rate. Finally, to avoid overfitting the network to the training data set, we employ

both a dropout rate and a $L^2$ regularization parameter to obtain a more robust network. We start with broad search ranges over the hyperparameters and assess the performance of the resulting model on the test set.

Based on numerous preliminary NOLH experimental designs, we find four layers to be sufficient to model the complexity of the data, employ a batch size of 256, and choose a $L^2$ regularization weight of $\lambda = 0.03$. We employ ReLu activation functions for the input layer and three hidden layers, and a softmax activation for the output to predict probabilities that a track falls into one of the nine classes. Fixing these parameters, we run a sequence of refined searches over the remaining four variables. Figure 3 shows the most refined experimental design used to choose the final network. This particular NOLH design operates by selecting a carefully chosen set of 17 design points that cover the space, while being nearly-orthogonal for statistical validity purposes. Thus, we feel comfortable that we have searched a space of possible combinations of hyperparameters to find meaningful interactions between options, rather than performing independent searches for each parameter on an *ad hoc* basis.

We finalize the remaining parameters by setting the number of neurons in each layer to 475, and the number of epochs to 288. Additionally, we set the dropout rate to 0.9%, and the learning rate to 0.003. This tuning results in 96.6% accuracy in classifying the training set, 88.9% accuracy on the validation set, and 68% unweighted F1-scores across the nine standard classes on the validation set. Some options tested from Figure 3 performed much worse, while others may have performed comparably in aggregate but at the expense of lower F1-scores in anomalous classes that do not have many samples. Our goal is to choose a model that will perform well on all classes, rather than simply optimizing for the one-way or Type 1 tracks. In the next section we report specific detailed results applied to the test set (which was not used at all in choosing the hyperparameters). The final network parameters are presented in Table 3.

The training process was relatively fast and completed within a few minutes on a personal computer. We built the network using the Keras API, and use Adam optimization with a

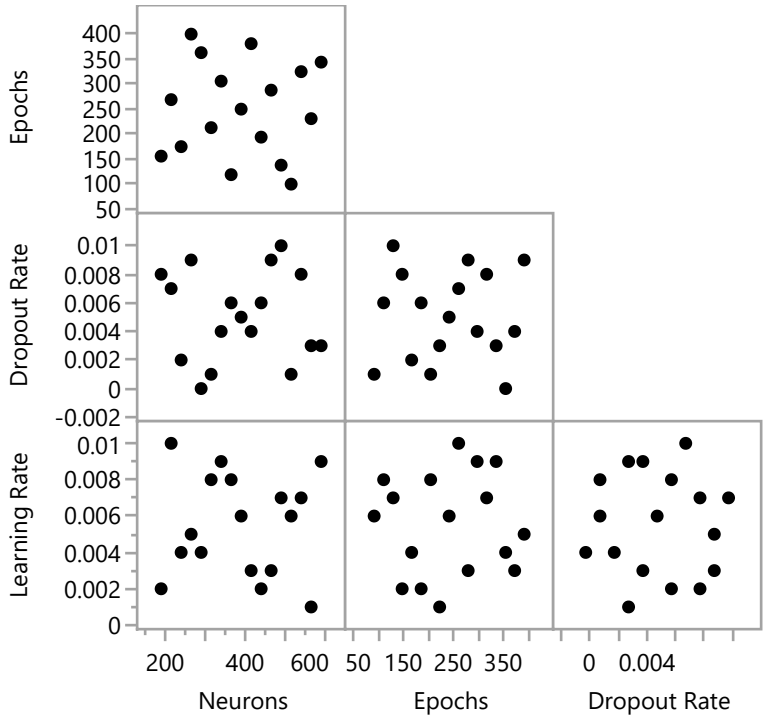



Figure 3: Refined hyperparameter tuning combinations tested for the number of neurons (width of the network), number of time epochs, the dropout rate, and the learning rate.

Table 3: Hyperparameter Values

| Hyperparameter | Value |
|---|---|
| Layers | 4 |
| Neurons/Layer | 475 |
| Learning Rate | 0.003 |
| Dropout Rate | 0.009 |
| $L^2$ Regularization | 0.03 |
| Batch Size | 256 |
| Epochs | 288 |

loss function of categorical crossentropy, i.e.,

$$Loss = -\sum_{i=1}^{N} w(c_i) \sum_{j=1}^{C} y_{i,j} \log(p_{i,j}),$$

where $N$ is the number of total observations in the validation set, $C$ is the number of categories, and $c_i$ is the correct label of observation $i$. For a given $i$, let $y_{i,j}$ be zero for all

$j$ except with a value of 1 at its correct label when $j = c_i$, and $p_{i,j}$ is the corresponding prediction probability that observation $i$ falls in category $j$. Additionally, we employ class weights inversely proportional to representation in the training sample size, represented by $w(c_i)$. This discourages overclassifying tracks according to the common shapes by providing higher weight to correct classification of the less frequent types.

# 4 Classification Results

We first present the results for the trained neural network on standard data in Section 4.1. Section 4.2 describes how the results of classifier trained on the standard data can be applied to the large volume of hybrid trajectories to identify anomalies.

## 4.1 Standard shape performance

First, we present the results for standard Types 1-9 using the trained classifier computed with the model developed in Section 3.3. Table 4 shows the precision, recall, and F1-score for each of the classes applied to the test set which was not used for training or tuning. Some shape types only have a few observations in the test set, so we also perform a 5-fold cross-validation on the entire standard data set and present the results in the right-most columns of Table 4.

Table 4: Category statistics for test set (left), and cross-validation results (right).

| | Test set | | | | 5-fold cross validation | | | |
|---|---|---|---|---|---|---|---|---|
| Class | Precision | Recall | F1 | Support | Precision | Recall | F1 | Support |
| Type 1 (straight) | 0.95 | 0.97 | 0.96 | 533 | 0.95 | 0.97 | 0.96 | 3435 |
| Type 2 (detour) | 0.68 | 0.59 | 0.63 | 58 | 0.68 | 0.59 | 0.63 | 375 |
| Type 3 (curved) | 0.89 | 0.71 | 0.79 | 66 | 0.84 | 0.82 | 0.83 | 501 |
| Type 4 (single loop) | 0.50 | 0.25 | 0.33 | 4 | 0.54 | 0.49 | 0.51 | 43 |
| Type 5 (multi-loop) | 1.00 | 0.50 | 0.67 | 2 | 0.67 | 0.53 | 0.59 | 15 |
| Type 6 (figure-eight) | 0.58 | 0.58 | 0.58 | 12 | 0.54 | 0.69 | 0.61 | 77 |
| Type 7 (out-back) | 0.20 | 0.25 | 0.22 | 4 | 0.56 | 0.43 | 0.49 | 46 |
| Type 8 (switchback) | 0.60 | 0.94 | 0.73 | 16 | 0.75 | 0.54 | 0.63 | 158 |
| Type 9 (sinusoidal) | 0.67 | 0.70 | 0.68 | 37 | 0.61 | 0.70 | 0.65 | 230 |

In general, we will focus the discussion on the cross-validation results since they are not as sensitive to the small sample sizes that occur for some of the classes in the test set. The classifier has varying performance across classes, with very strong performance for Type 1 (straight line) with an F1-score of 0.96 for both the test set and cross validation results. This class also comprises 70% of the data set, so the overall performance of the classifier will be high because of the predominance of Type 1 trajectories. Type 3 (curved) is the next most common type, and also performs relatively well with an F1-score of 0.83. Type 2 (single detour), Type 8 (switchback) and Type 9 (sinusoidal) are the next most observed types and have comparable F1-scores of 0.63, 0.63, and 0.65 respectively. Presumably, it is easier for the model to train on types with more observations. Type 4 (single loop), Type 5 (multi-loop), Type 6 (figure-eight) and Type 7 (out and back) have lower F1-scores, potentially due to infrequent observations, but also for additional reasons considered below related to potential misclassification between types.

Table 5 displays the aggregate performance of the classifier applied to the test set, and the cross-validation results. The unweighted averages are the simple averages calculated across the columns in Table 4 while the weighted averages are weighted according to the frequency of the classes. Because of the high frequency of Type 1, the weighted F1-scores are 0.88 for the test set and cross-validation results. However, the unweighted results across classes reveal an F1-score of 0.62 for the test set, and 0.66 in the cross validation. This is still significantly better than a random chance classification across the nine types, however, obtaining more observations to train the less common types would potentially improve performance in these classes.

Table 5: Aggregate Performance Metrics.

| | Test set | | 5-fold cross validation | |
|---|---|---|---|---|
| Metric | Unweighted Avg | Weighted Avg | Unweighted Avg | Weighted Avg |
| Precision | 0.67 | 0.89 | 0.68 | 0.88 |
| Recall | 0.61 | 0.89 | 0.64 | 0.88 |
| F1 Score | 0.62 | 0.88 | 0.66 | 0.88 |

While Tables 4 and 5 suggest that the classifier will perform significantly better than a random chance draw across nine possible categories, Figure 4 displays the normalized confusion matrix for the test set and cross validation results which yields more insight into the nature of the misclassifications. We see that many of the Type 2 (detour) tracks are misclassified as Type 1, which makes sense because they are mostly straight aside from a single turn. The Type 3 (curved) tracks may be mistaken as straight or detours. The fact that there may be confusion among Types 1, 2 and 3 is not surprising because they all travel generally one-way from one location to another. Type 9 (sinusoidal/jagged) is like Type 2 but with multiple turns, so also has a relatively high misclassification as Type 1. Type 8 (switchback) is mostly straight, so some will be misclassified as Type 1, but otherwise performs quite well.

One additional source of misclassification between Types 1-3 comes from the way the display projection affected the manual labeling. The authors used the simple equidistant cylindrical map projection, where latitude and longitude are directly converted to Cartesian coordinates to manually classify the tracks. For long international flights this projection may distort the true trajectory shape. For example, a long curved trajectory of Type 3 may correspond to a transatlantic flight that likely flies close to the straight great-circle path. Thus this type of track might be more appropriately classified as Type 1. This distortion will degrade the performance of our algorithm. However, this issue only impacts a relatively small number of trajectories and will primarily affect Types 1, 2, and 3 because longer flights are likely one of those types.

We also can explain the intuition behind the misclassification of the unusual curved shape types. Types 4 through 7 all have endpoints close to the starting point, so there is some misclassification between these classes, but not as much between these and Types 1-3. Type 5 (multi-loop) may be classified as Type 4 (single loop), or Type 6 (a figure-eight) due to high curvature. Type 4 might be similar to a Type 6 or Type 7 (out-back) in that it returns to the starting point once. The difference between Type 6 and Type 7 is mainly that there

is an intersection in the path in figure-eights, but not in out-backs, so there will be some misclassifications between the two.

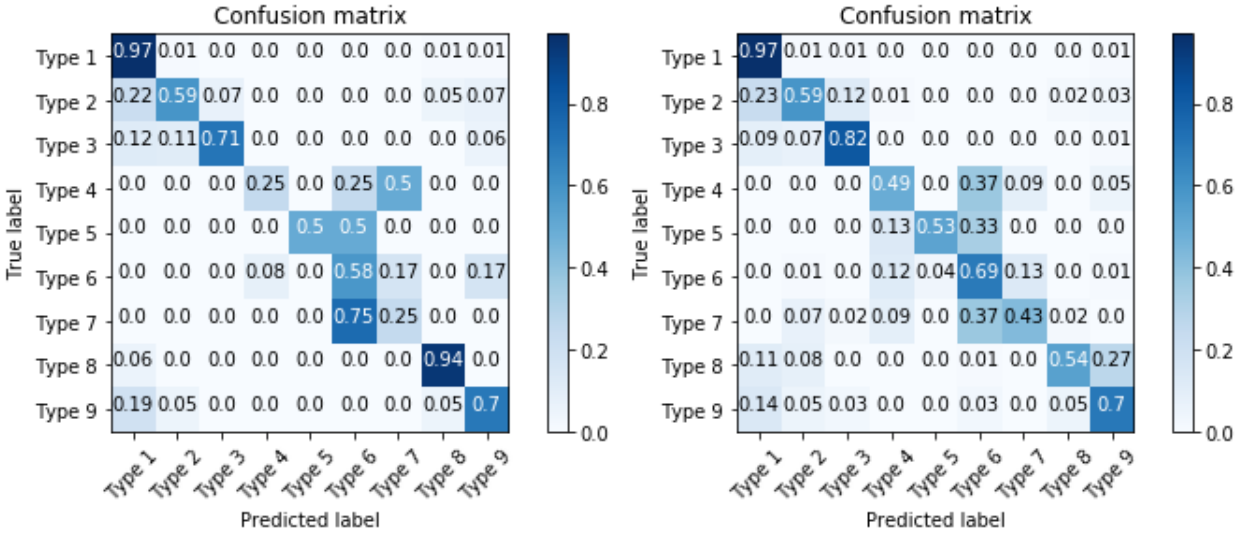



Figure 4: Normalized confusion matrix for test set (left) and cross validation (right).

Generally, we find the classifier can separate out straight tracks (Type 1) with very high precision and recall. It can also differentiate those that travel in a mostly straight-forward manner between two start and end points from those that return to their starting point. There is very little misclassification between Types 4-7 and the other types. The classifier appears promising in being able to differentiate between different types of curvature, but may require more specific criteria than the input variables we used to be able to differentiate the more complicated shapes from each other with high accuracy. Additionally, the tracks reveal high variability relative to the ideal form the shapes would take. The subjective error in human labeling is mainly the result of this variability, so there are natural limits to what we expect the model to predict based on large uncertainty across trajectories that could be classified as the same shape.

## 4.2 Hybrid shape performance

As most of the tracks were placed in the hybrid class due to the high variation in flight trajectories, we study the effect of applying the trained classifier from the standard data to the hybrid data. Because much of the hybrid data was *hybrid-1way* (H1) in that it appeared as a standard one-way trip with some variation from Types 1-3, we expect that the classifier trained on standard types may still be able to detect anomalous trajectories similar to Types 4-9. We call those hybrid trajectories that don't appear similar to Types 1-3 as *hybrid-anomalous* (HA), see Table 1 for a breakdown.

Let $c_i$ be the true label of trajectory $i$. This label can be either an integer 1 through 9 if the shape is standard, or it can be classified as H1 or HA. The classifier can be applied to a trajectory and will deliver a predicted class $\hat{c}_i$ which can be 1 through 9. For each track, the prediction is a result of the softmax output layer of the network, where the probability that track $i$ is shape $j$ is computed as $p_{i,j}$ with $\hat{c}_i = \text{argmax}_j p_{i,j}$. The softmax output vector $p_{i,j}, j = 1, \ldots, 9$ contains information on which shape types are most likely for a track, with $\sum_{j=1}^{9} p_{i,j} = 1$. We first look at the overall rate that the classifier assigns tracks to shapes across different data subsets.

Table 6: Probability distribution for predictions across classes.

| Type | Training | Validation | Test | H1 | HA |
|---|---|---|---|---|---|
| | True Frequency (%) | Predicted Frequency (%) | | | |
| Type 1 (straight) | 69.17 | 74.18 | 72.95 | 47.28 | 13.56 |
| Type 2 (detour) | 7.85 | 5.05 | 5.74 | 17.71 | 14.75 |
| Type 3 (curved) | 10.77 | 9.84 | 9.97 | 15.56 | 13.60 |
| Type 4 (single loop) | 0.91 | 0.41 | 0.41 | 0.01 | 1.29 |
| Type 5 (multi-loop) | 0.32 | 0.14 | 0.14 | 0.01 | 0.70 |
| Type 6 (figure-eight) | 1.64 | 1.64 | 1.91 | 0.11 | 3.87 |
| Type 7 (out-back) | 1.05 | 0.96 | 0.41 | 0.01 | 0.45 |
| Type 8 (switchback) | 3.34 | 3.14 | 2.60 | 3.62 | 12.36 |
| Type 9 (sinusoidal) | 4.95 | 4.64 | 5.87 | 15.69 | 39.42 |

Table 6 compares the breakdown of classifications of trajectories across different data sets. First, the labeled training set for standard tracks is broken down across classes, where we

observe most standard tracks (69.17%) are Type 1. We then show the breakdown of predicted classes for the validation and test sets. Since these sets should have the same overall distribution of data due to being drawn from the standard data set, they reveal similar breakdowns. A $\chi^2$-goodness-of-fit test reveals a $p$-value of 0.33 when comparing the prediction distributions between the validation and test prediction distributions. However, the validation and test sets have significantly different distributional breakdowns from the training and H1 and HA data sets ($\chi^2$-test $p$-value of 0). This means the classifier is likely biased, for example, more tracks may be assigned to Type 1 relative to their true proportion (69.17%) in the labeled data, with 74.18% and 72.95% for the validation and test sets respectively. Unsurprisingly, the breakdowns for H1 and HA will be different from those of the validation and test sets because the properties of the underlying data is fundamentally different.

We next study the columns for H1 and HA in Table 6. In the H1 column, we see that a majority of the tracks are assigned to Types 1-3 (80.55%), which is reassuring because the one-way trips that did not strictly meet the criteria to be classified as standard will still be labeled as one-way. Of the remaining tracks, 15.69% were classified as Type 9, possibly if there were multiple turns. Only 0.14% were classified as Types 4-7, which means that even if a track cannot be outright classified as Types 1-3, it is unlikely to be misclassified as a return trip if is somewhat similar to Types 1-3. On the other hand, the HA data set has far fewer tracks classified as Types 1-3 (41.91%), but this still implies the classifier is likely to have a high false negative rate in assigning tracks that the human viewed as anomalous as one-way. However, there was a much higher number of tracks assigned to Types 4-9 than in any of the other data sets. This analysis reveals that the classifier may still be able to identify truly anomalous trajectories even when the track does not fit nicely into one of the standard types, at the expense of accidentally classifying anomalous tracks as Types 1-3.

We can also look at the confidence of the prediction as a measure of quality of the classifier applied to hybrid data. For a given track $i$, the most focused prediction would put probability 1 in one value of $p_{i,j}$ and 0 for the rest of the shapes $j$, while a random chance guess would

put probability 1/9 for each $p_{i,j}$. Table 7 reports metrics on the average strength levels of the prediction for each class. Let $N$ be the total number of data samples in a prediction set, and $N_j$ be the number of trajectories assigned to class $j$ from that particular prediction set where $\mathcal{I}$ is the indicator function so

$$N_j = \sum_{i=1}^{N} \mathcal{I}(\hat{c}_i = j).$$

Then the average maximum prediction for a given class $j$ is

$$\frac{1}{N_j} \sum_{i=1}^{N} \mathcal{I}(\hat{c}_i = j) p_{i,j}, \tag{1}$$

where if $\mathcal{I}(\hat{c}_i = j)$ is true then $p_{i,j} = \max_k p_{i,k}$. Table 7 presents these average maximum prediction values using (1). For the classifier applied to the standard test set data, we see the average prediction strengths are quite high, meaning that the classifier is on average giving a preponderance of weight to one class when making a prediction. These values are much lower for H1 than for the test data, meaning that the classifier is not placing these H1 trajectories as strongly in any one class. This makes sense since these tracks do not obviously fit into any of the shapes, and most of them, while appearing one-way, are not clearly in Types 1-3 according to the human classifier. The HA data set also has lower average prediction strengths for Types 1-5 and 7. For Type 6, 8, and 9, the average prediction strength is slightly higher for the HA data, meaning the classifier may strongly associate these anomalous tracks with some of the anomalous shape types.

Finally, we consider the entropy of the probability predictions for a given track $i$. These softmax predictions are the values $p_{i,j}, j = 1, \ldots, 9$, and the entropy of a distribution is a measure of dispersion. Entropy for a prediction distribution with all its mass at one point is 0, while maximum entropy is achieved for the discrete uniform distribution. Entropy for

Table 7: Average maximum prediction probabilities, and average entropy of predictions (bottom row).

| Type | Standard (Test) | H1 | HA |
|---|---|---|---|
| Type 1 (straight) | 94.7 | 84.4 | 79.4 |
| Type 2 (detour) | 78.1 | 72.6 | 76.2 |
| Type 3 (curved) | 86.8 | 78.1 | 81.2 |
| Type 4 (single loop) | 85.1 | 61.0 | 67.3 |
| Type 5 (multi-loop) | 99.6 | 94.1 | 92.2 |
| Type 6 (figure-eight) | 75.8 | 64.7 | 79.1 |
| Type 7 (out-back) | 75.9 | 38.8 | 65.2 |
| Type 8 (switchback) | 78.9 | 76.2 | 79.2 |
| Type 9 (sinusoidal) | 78.5 | 76.0 | 82.7 |
| Avg Prediction Entropy | 0.283 | 0.562 | 0.539 |

track $i$ can be calculated as

$$-\sum_{j=1}^{9} p_{i,j} \log p_{i,j}.$$

The bottom row of Table 7 calculates the average entropy for all prediction probability vectors in the data set. We see that for the standard test data the entropy is 0.283, which is much lower than those for H1 and HA which are 0.562 and 0.539. This suggests that in addition to the maximum probability assignments being lower on average for hybrid tracks, the probability dispersion across shapes will be higher. This means that the strength of the maximum probability prediction and dispersion levels may be used to identify anomalous tracks that do not fit strongly into any of the shape types. These metrics can separate tracks that strongly match the standard types from those that may be in the hybrid class.

We conclude with some illustrative examples of highly anomalous trajectories which are classified by the neural network. In Figure 5 we consider two trajectories that could be close to figure-eights. The left one is classified as figure-eight with probability 89%, and a second choice of sinusoidal with probability 8.8%, with minimal weights given to loop and out-back. The overall entropy of the classification is 0.418. Understandably, the classifier is affected by the changes in direction and curvature of the track. The plot on the right is classified as a switchback with probability 83.1%, and sinusoidal with 9.6%, with minimal assignments to

detour, loop, figure-eight, and out-back. The overall entropy is higher at 0.640. The start and end points are far away from each other, and there are multiple sharp changes in direction, which explains why switchback and sinusoidal are chosen as most likely.

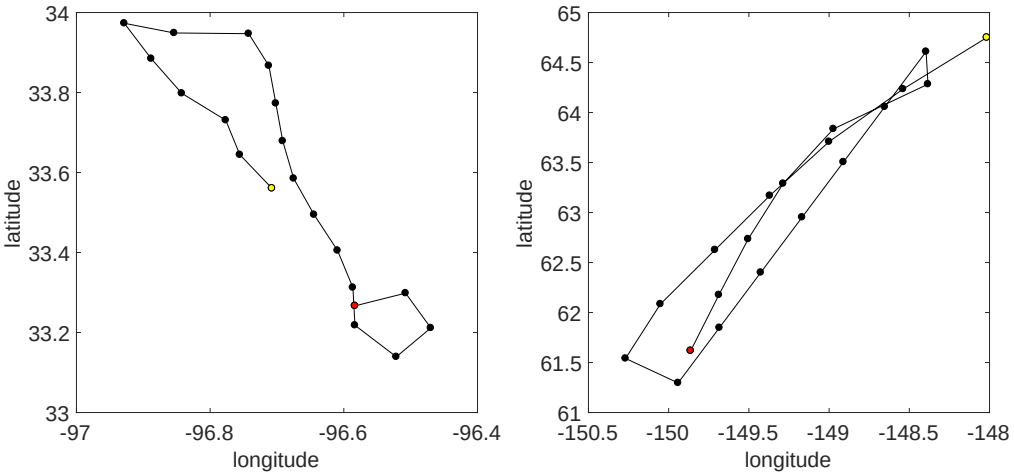



Figure 5: Left: Classified as Type 6 (figure-eight) with probability 89.0% and entropy 0.418. Right: classified as switchback (Type 8) with probability 83.1% and entropy 0.640.

Figure 6 has two figures which are mostly parabolic in shape, with some significant anomalies. The left plot is classified as sinusoidal with probability 96.3%, with smaller probabilities assigned to detour, curved, loop, figure-eight, and switchback. However, the high weight on sinusoidal drives the entropy down to 0.198. The distance between the start and end point rules out Types 4-7, but the extra curvature and changes in direction for the mini-loop correctly prevent the track from being classified as Type 3. The right plot assigns the track as a figure-eight with probability 92.4%, with the remainder of the weight on loop, out-back, and sinusoidal. The overall entropy is 0.341, and the track does display properties of a figure-eight by going out and crossing over itself on the way back. It is possible that the classifier can be used for feature extraction, and while it cannot perform well on highly unusual shapes such as these for which it is untrained, it may be able to spot them as anomalies (i.e, not Type 1-3).

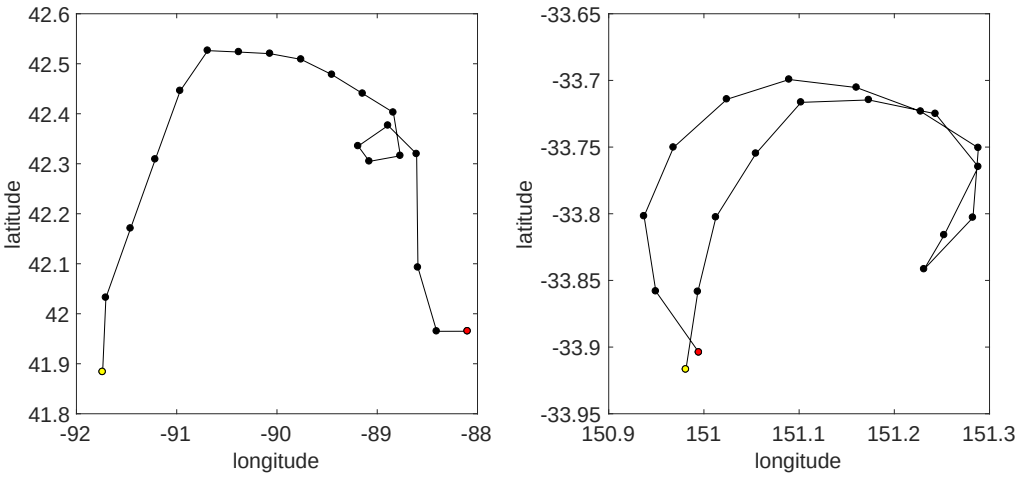



Figure 6: Left: Classified as sinusoidal with probability 96.3% and entropy 0.198. Right: classified as figure-eight with probability 92.4% and entropy 0.341.

# 5 Conclusions

This research represents a first attempt to classify shapes of flight trajectories from ADS-B data using neural networks. Given the large volume of ADS-B data available, it appears possible to train a classifier using human labeled samples to classify a large range of ADS-B trajectories. We build a neural network using as inputs common shape features such as change in heading and curvature, and also consider the distribution between different points in the track that may differentiate shapes. The advantage of using neural networks is that it is possible to quickly train and capture nonlinear relationships using a large number of complex inputs, and thus it is possible to save human effort by identifying potentially anomalous behavior automatically. Because of the high variability present in the shapes of the trajectories (as seen through high error when multiple novices label the same data), we see potential benefits of using machine learning to filter out common shape types.

We train the neural network on a clean dataset of labeled standard trajectories consisting of nine shape types. In order to tune the network, we employ a design of experiments to explore the space of possible model parameters. The resulting model performs reasonably

well on previously unseen test data, with an overall weighted F1-score of 0.88, and an unweighted average across the nine classes of 0.62, implying the model is reasonably robust across different trajectory types. Cross-validation efforts over the entire standard data set deliver similar results. We also apply the classifier to trajectories that do not fit cleanly into the nine main shape types (hybrid types), and demonstrate the classifier may still be able to isolate anomalous trajectories from those that are mostly straight or one-way for these types of unclassified data. The model will likely provide stronger predictions for trajectories more closely meeting the standard shape types, either through high maximum probability assignments or lower entropy.

The classifier does a good job separating one-way from return trips, and shapes with minimal curvature from those with large amounts of curvature. Future efforts could spend more time closely defining anomalous shapes of interest and collecting or simulating more types of these tracks to provide additional data for the model to train on. Furthermore, we did not consider the location or altitude of the track, and these are features that may correlate with the shape. We also used a small sample of location values (only 20 points per track) for the change in heading and curvature values which allowed for each trajectory to be represented by the same number of input features. However, using more of the information available in each track may yield potentially useful inputs, or allow von Mises distributions to be fit on sequences of angles. Given the robustness of the model presented on the inputs chosen and the speed of training the network, there are numerous other inputs that could be used to identify specific anomalies, and neural networks continue to be a promising tool for quickly automating the classification process.

## Acknowledgements

We are extremely grateful to the Center for Multi-Intelligence Studies at the Naval Postgraduate School for their support of this work.

# References


[1] Sillito, R. R. and Fisher, R. B., "Semi-supervised learning for anomalous trajectory detection." *BMVC*, Vol. 1, 2008, pp. 035–1.

[2] Junejo, I. N., Javed, O., and Shah, M., "Multi feature path modeling for video surveillance," *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 2, IEEE, 2004, pp. 716–719.

[3] Morris, B. T. and Trivedi, M. M., "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 8, 2008, pp. 1114–1127.

[4] Georgiou, H., Karagiorgou, S., Kontoulis, Y., Pelekis, N., Petrou, P., Scarlatti, D., and Theodoridis, Y., "Moving objects analytics: Survey on future location & trajectory prediction methods," *arXiv preprint arXiv:1807.04639*, 2018.

[5] Nguyen, D.-D., Le Van, C., and Ali, M. I., "Vessel trajectory prediction using sequence-to-sequence models over spatial grid," *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, 2018, pp. 258–261.

[6] Nawaz, T., Cavallaro, A., and Rinner, B., "Trajectory clustering for motion pattern extraction in aerial videos," *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, pp. 1016–1020.

[7] Boyle, J., Nawaz, T., and Ferryman, J., "Deep trajectory representation-based clustering for motion pattern extraction in videos," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2017, pp. 1–6.

[8] Smart, E., Brown, D., and Denman, J., "A two-phase method of detecting abnormalities in aircraft flight data and ranking their impact on individual flights," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 3, 2012, pp. 1253–1265.

[9] Matthews, B., Das, S., Bhaduri, K., Das, K., Martin, R., and Oza, N., "Discovering anomalous aviation safety events using scalable data mining algorithms," *Journal of Aerospace Information Systems*, Vol. 10, No. 10, 2013, pp. 467–475.

[10] Li, L., Das, S., John Hansman, R., Palacios, R., and Srivastava, A. N., "Analysis of flight data using clustering techniques for detecting abnormal operations," *Journal of Aerospace Information Systems*, Vol. 12, No. 9, 2015, pp. 587–598.

[11] Chidester, T. R., "Understanding normal and atypical operations through analysis of flight data," *Proceedings of the 12th International Symposium on Aviation Psychology*, Citeseer, 2003, pp. 239–242.

[12] Li, L., Gariel, M., Hansman, R. J., and Palacios, R., "Anomaly detection in onboard-recorded flight data using cluster analysis," *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, IEEE, 2011, pp. 4A4–1.

[13] Li, L., *Anomaly detection in airline routine operations using flight data recorder data*, Ph.D. thesis, Massachusetts Institute of Technology, 2013.

[14] Li, L., Hansman, R. J., Palacios, R., and Welsch, R., "Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring," *Transportation Research Part C: Emerging Technologies*, Vol. 64, 2016, pp. 45–57.

[15] Li, G., Lee, H., Rai, A., and Chattopadhyay, A., "Analysis of operational and mechanical anomalies in scheduled commercial flights using a logarithmic multivariate Gaussian model," *Transportation Research Part C: Emerging Technologies*, Vol. 110, 2020, pp. 20–39.

[16] Gariel, M., Srivastava, A. N., and Feron, E., "Trajectory clustering and an application to airspace monitoring," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 4, 2011, pp. 1511–1524.

[17] Olive, X. and Basora, L., "Identifying anomalies in past en-route trajectories with clustering and anomaly detection methods," Tech. Rep. hal-02345597, VIENNE, Austria, 2019.

[18] Song, Y., Cheng, P., and Mu, C., "An improved trajectory prediction algorithm based on trajectory data mining for air traffic management," *2012 IEEE International Conference on Information and Automation*, IEEE, 2012, pp. 981–986.

[19] Evans, A. D., Lee, P., and Sridhar, B., "Predicting the operational acceptance of airborne flight reroute requests using data mining," *Transportation Research Part C: Emerging Technologies*, Vol. 96, 2018, pp. 270–289.

[20] Puranik, T. G. and Mavris, D. N., "Anomaly detection in general-aviation operations using energy metrics and flight-data records," *Journal of Aerospace Information Systems*, Vol. 15, No. 1, 2018, pp. 22–36.

[21] Deshmukh, R. and Hwang, I., "Incremental-Learning-Based Unsupervised Anomaly Detection Algorithm for Terminal Airspace Operations," *Journal of Aerospace Information Systems*, Vol. 16, No. 9, 2019, pp. 362–384.

[22] Le Fablec, Y. and Alliot, J.-M., "Using neural networks to predict aircraft trajectories." *IC-AI*, 1999, pp. 524–529.

[23] Cheng, T., Cui, D., and Cheng, P., "Data mining for air traffic flow forecasting: A hybrid model of neural network and statistical analysis," *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, Vol. 1, IEEE, 2003, pp. 211–215.

[24] Kulkarni, V. B., "Intelligent air traffic controller simulation using artificial neural networks," *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, IEEE, 2015, pp. 1027–1031.

[25] Memon, Q., "Intelligent control of air traffic landing sequences," *International Journal of Modelling and Simulation*, Vol. 28, No. 1, 2008, pp. 35–42.

[26] Liu, H., Lin, Y., Chen, Z., Guo, D., Zhang, J., and Jing, H., "Research on the air traffic flow prediction using a deep learning approach," *IEEE Access*, Vol. 7, 2019, pp. 148019–148030.

[27] Shi, Z., Xu, M., Pan, Q., Yan, B., and Zhang, H., "LSTM-based flight trajectory prediction," *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

[28] Anjum, N. and Cavallaro, A., "Multifeature object trajectory clustering for video analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 11, 2008, pp. 1555–1564.

[29] Zaki, M. H. and Sayed, T., "A framework for automated road-users classification using movement trajectories," *Transportation Research Part C: Emerging Technologies*, Vol. 33, 2013, pp. 50–73.

[30] Murça, M. C. R., Hansman, R. J., Li, L., and Ren, P., "Flight trajectory data analytics for characterization of air traffic flows: A comparative analysis of terminal area operations between New York, Hong Kong and Sao Paulo," *Transportation Research Part C: Emerging Technologies*, Vol. 97, 2018, pp. 324–347.

[31] Tao, Y., Faloutsos, C., Papadias, D., and Liu, B., "Prediction and indexing of moving objects with unknown motion patterns," *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 2004, pp. 611–622.

[32] Guo, Y., Xu, Q., Yang, Y., Liang, S., Liu, Y., and Sbert, M., "Anomaly detection based on trajectory analysis using kernel density estimation and information bottleneck techniques," *Tech. Rep., Technical Report 108*, University of Girona, 2014.

[33] Guo, Y., Xu, Q., Li, P., Sbert, M., and Yang, Y., "Trajectory shape analysis and anomaly detection utilizing information theory tools," *Entropy*, Vol. 19, No. 7, 2017, pp. 323.

[34] Prati, A., Calderara, S., and Cucchiara, R., "Using circular statistics for trajectory shape analysis," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[35] Calderara, S., Prati, A., and Cucchiara, R., "Mixtures of von Mises distributions for people trajectory shape analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 21, No. 4, 2011, pp. 457–471.

[36] Mcfadyen, A., O'Flynn, M., Martin, T., and Campbell, D., "Aircraft trajectory clustering techniques using circular statistics," *2016 IEEE Aerospace Conference*, IEEE, 2016, pp. 1–10.

[37] Gingrass, C., *Classifying ADS-B Trajectory Shapes Using a Dense Feed-Forward Neural Network.*, Master's thesis, Naval Postgraduate School, 2020.

[38] ADS-B Exchange, 2020.

[39] The OpenSky Network, 2020.

[40] Sun, J., Ellerbroek, J., and Hoekstra, J., "Flight extraction and phase identification for large Automatic Dependent Surveillance–Broadcast datasets," *Journal of Aerospace Information Systems*, Vol. 14, No. 10, 2017, pp. 566–572.

[41] Williams, E., "Aviation Formulary V1.46," 2011.

[42] Léger, J.-C., "Menger curvature and rectifiability," *Annals of Mathematics*, Vol. 149, 1999, pp. 831–869.

[43] Sanchez, S., "NOLHdesigns spreadsheet," 2011.

[44] Cioppa, T. M. and Lucas, T. W., "Efficient nearly orthogonal and space-filling Latin hypercubes," *Technometrics*, Vol. 49, No. 1, 2007, pp. 45–55.