Re: BREAKING NEWS: PARANORMAL ENTITIES ARE TAKING PILOTS' JOBS
From: AlienInvasionPreventionSquad@area51.com

Hello Brian from the CIA,

Thank you for your inquiry regarding the increasingly anomalous behavior of flights across the United States of America as well as for providing the *Flights1_2019_1* dataset. Our explorations confirm that several airports are showing increasingly strange behavior and may have been taken over by the foreign entity invasion. Please see the report below for the complete investigation details.

Variable	Explanation	Example
1: YEAR	Year at which flight departed (every entry has value 2019)	2019
2: DAY_OF_WEEK	Day of week flight departed represented as value 1-7 (Monday: 1, Sunday: 7)	3
3: FL_DATE	Date at which the flight departed	2019-01-25
4: ORIGIN_AIRPORT_ID	Unique identifier that represents an airport over a large period of time (departure)	13487
5: ORIGIN_AIRPORT_SEQ_ID	Unique "sequence" identifier given to each airport over a shorter duration of time (departure)	1348702
6: ORIGIN_CITY_MARKET_ID	Unique identifier given to each airport representing the airport's "market" (departure)	31650
7: ORIGIN_CITY_NAME	Name of departure city, along with abbreviated state city belongs to	Minneapolis, MN
8: DEST_AIRPORT_ID	Unique identifier given to each airport to represent airport over large period of time (arrival)	13487
9: DEST_AIRPORT_SEQ_ID	Unique "sequence" identifier given to each airport over shorter duration of time (arrival)	1348702
10: DESR_CITY_MARKET_ID	Unique identifier given to each airport representing the airport's "market" (arrival)	31650
11: DEST_CITY_NAME	Name of arrival city, along with abbreviated state city belongs to	Minneapolis, MN
12: DEST_STATE_ABR	Abbreviation of state which flight arrives to	MN
13: DEP_DELAY	Departure delay from scheduled, shown in minutes (may be negative if flight departs before scheduled)	17
14: ARR_TIME	Time at which flight arrives represented in form hhmm (24 hour time)	1947
15: ARR_DELAY	Arrival delay from scheduled, shown in minutes (may be negative if flight arrives before scheduled)	19
16: ARR_DELAY_NEW	Represents arrival delay when flight arrives after scheduled (ARR_DELAY is positive), else takes on value of 0	8
17: ARR_DEL15	Flag showing when ARR_DELAY is greater than or equal to 15 (1 if >=, else 0)	{0,1}

Table 1. Data Dictionary for Flights1_2019_1 dataset

We investigated potential anomalies in two scopes, by looking for abnormalities within the various values of a record and by contrasting records against the entire dataset. To do the latter, we started by first confirming that there were no intra-record anomalies.

For example, do the various airport identifiers coincide with each other? Do the various measurements for flight date and time match? Do the manipulations of the flight delays behave in the way they were set to? And, as seen in Figure 1, do the average departure and arrival delays reasonably follow each other? (They do). Investigating these alongside many other relationships within each record allowed us to explore the dataset however uncovered nothing abnormal except, a small hiccup relating to several airport states not matching the state of the city the airport was marked to. However these exceptions were explained by the cities in question bordering another state, in which its airport lay. After verifying these rules, we could then proceed exploring and transforming our data.

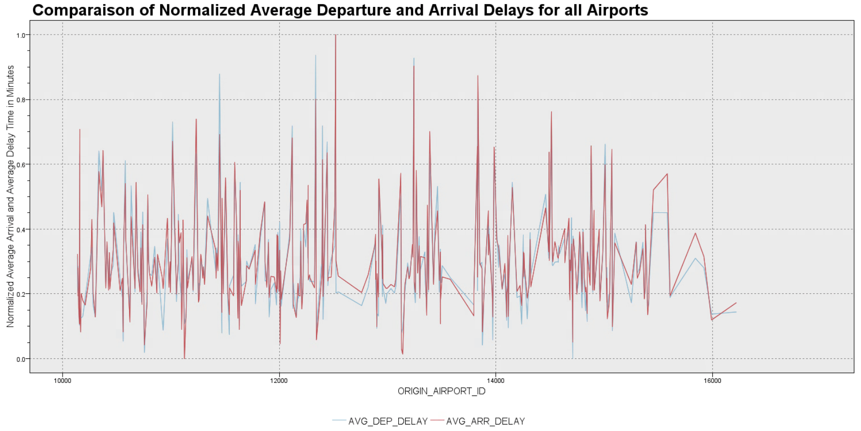


Figure 1. An example of intra-record outlier validation.

The full scope of our intra-variable relationship validation is as follows. All relationships were confirmed as true for the dataset

- Because every observation has “2019” for the year variable, we should also expect the year term in the FL_DATE variable to also always have a value of 2019. This is true for all records.
- The day of the week shown in V2 should align with the date in the FL_DATE variable. This was checked using the lubridate package in R and is true for all variables.
- For each departure and arrival airport, their ID and SEQ ID values should remain consistent with each other. This is true for all variables.
- The state which is abbreviated in V12 should match with that of V11.
- Departure and arrival delays will not translate 1:1, however we may logically expect departure delay to have a correlation with the arrival delay. They do.
- Because the variable ARR_TIME is in the form hhmm, the hh component must take values 00-24, whereas the mm component will take values 00-59. This is true for all values.
- Omitting rows containing NA values, ARR_DELAY_NEW behaves as outlined in the variable explanation.
- The ARR_DELAY15 Index behaves in line with ARR_DELAY.

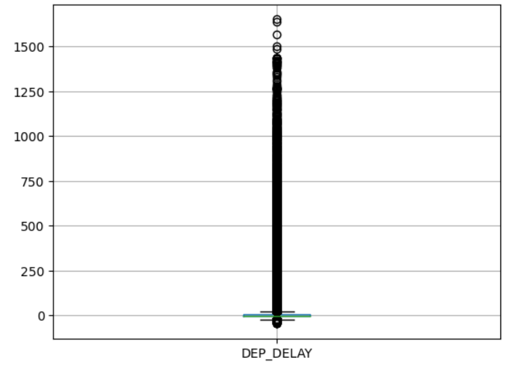


Figure 2a. Boxplot of DEP_DELAY

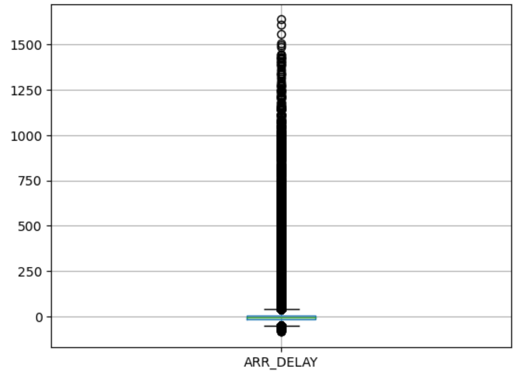


Figure 2b. Boxplot of ARR_DELAY

In an attempt to find outliers from the given data, we created boxplots of the departure delay (Figure 2a) and arrival delay (Figure 2b) variables. As expected, many flights depart and arrive on time, so the median is 0, and many flights are close to being on time, so the interquartile range is small. There are 80032 and 50685 outliers (points outside of the interquartile range) for DEP_DELAY and ARR_DELAY, respectively. This is too many data points to draw conclusions about anomalies. The overall enormous volume of data led us to derive a new dataset from the Flights1_2019_1 dataset, grouped by ORIGIN_AIRPORT_ID. This new dataset contains 346 rows (unique origin airport IDs) and 19 columns. We created columns that we thought summarized the data of each origin airport ID well. They are described in the table below.

Variable	**Explanation**	**Example**
ORIGIN_AIRPORT_ID	Unique identifier given to each airport to represent airport over large period of time (departure)	13487
AVG_DEP_DELAY	Average departure delay in minutes	12.209
VAR_DEP_DELAY	Variance within departure delay minutes	2998.078
MAX_DEP_DELAY	Maximum departure delay in minutes	-15
MIN_DEP_DELAY	Minimum departure delay in minutes	1348702
OUTLIERS_DEP_DELAY	Number of outliers in the departure delay (observations below Q1 − 1.5 IQR or above Q3 + 1.5 IQR)	43
AVG_ARR_TIME	Average arrival time in minutes	1395.97
VAR_ARR_TIME	Variance within arrival time in minutes	251233
MAX_ARR_TIME	Maximum time at which flight arrives represented in form hhmm (24 hour time)	2314
MIN_ARR_TIME	Minimum time at which a flight arrives represented in form hhmm (24 hour time)	3
AVG_ARR_DELAY	Average arrival delay in minutes	8.22
VAR_ARR_DELAY	Variance in arrival delay in minutes	3027.42
MAX_ARR_DELAY	Maximum arrival delay in minutes	331

MIN_ARR_DELAY	Minimum arrival delay in minutes	-48
OUTLIERS_ARR_DELAY	Number of outliers in the departure delay (observations below Q1 − 1.5 IQR or above Q3 + 1.5 IQR)	33
PROPORTION_COUNT_DEP_DELAY_10_OR_MORE	Number of departure delays greater than 10 minutes divided by number of departures at that airport *Note:* 10 minutes is the average departure delay across all entries of the original dataset	0.193
PROPORTION_OUTLIERS_DEP_DELAY	Number of outlier departure delays divided by number of departures at that airport	0.133
PROPORTION_COUNT_ARR_DELAY_5_OR_MORE	Number of arrival delays greater than 5 minutes divided by number of arrivals at that airport *Note:* 5 minutes is the average arrival delay across all entries of the original dataset	0.299
PROPORTION_OUTLIERS_ARR_DELAY	Number of outlier arrival delays divided by number of arrivals at that airport	0.1028

Table 2. Data dictionary for derived dataset

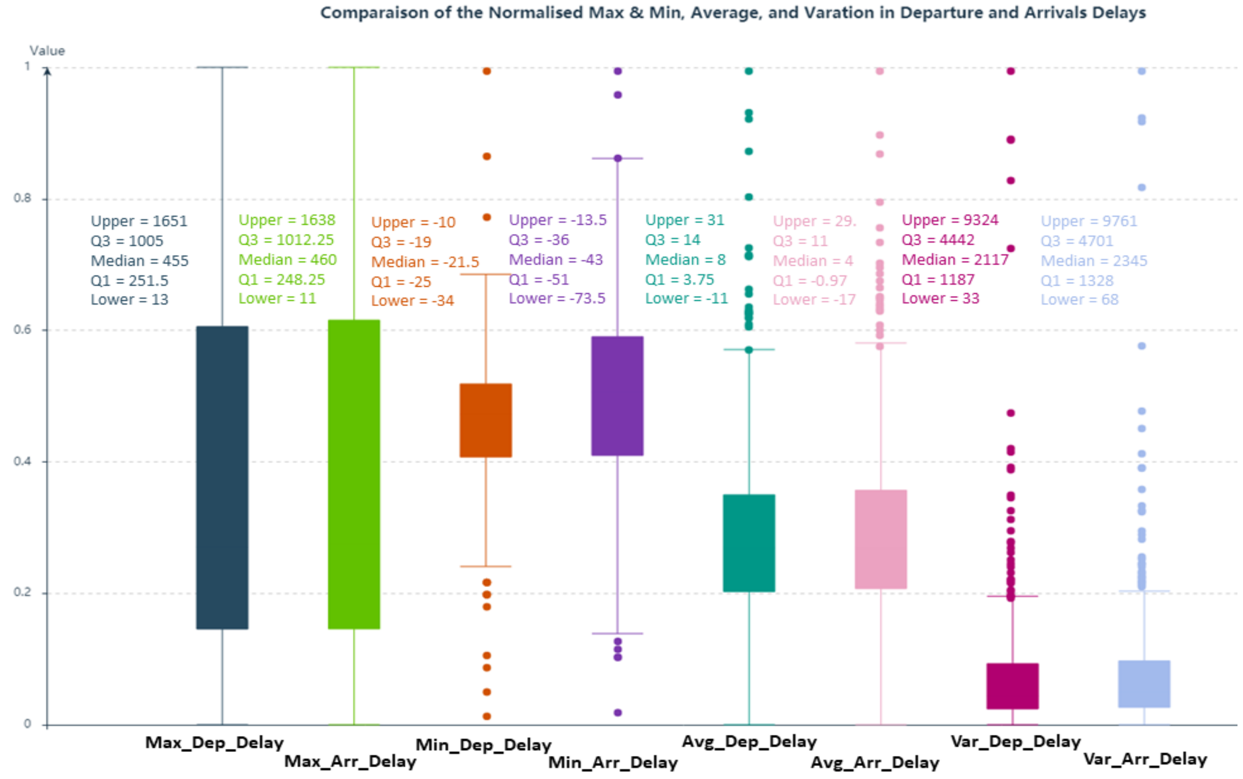


Figure 3. Summary Statistics about our reduced datasets key variables

After reducing our original *58398 row by 16 column dataset* to a *346 row by 19 column dataset*, we shrank it even further to a *346 row by 5 column dataset* using a Principal Component Analysis dimensionality reduction algorithm that would keep all of the most important information and patterns. We used 5 components in this process because an analysis of “the importance of components” showed that 5 led to a cumulative proportion of about 96.78% of the variance in the data being explained. Using this new PCA obtained dataset, we were able to run algorithms on the dataset to detect outliers within individual records in relation to the dataset as a whole.

Our Anomaly detection algorithms

Our first attempt at anomaly detection on the PCA reduced dataset was to look at distance-based methods. We looked at distances from each point in the dataset (the airport) to other points. Points who have large distances from other points are identified as outliers. Since we are working with a 5-dimensional space, we used distance measures Mahalanobis distance, Euclidean distance, Chebychev distance, and Manhattan distance. These are distance measures that are able to capture the relationship between the points and reduce to a scalar distance value. The Mahalanobis distance between two points **p** and **q** is $\sqrt{(p - q)^T \Sigma^{-1} (p - q)}$, where Σ^{-1} is the inverse of the covariance matrix, and superscript 'T' is the transpose of the vector (**p** - **q**). Figure 4a captures, for each observation (airport), the mean Mahalanobis Distance to all other points (airports). Observations with higher mean Mahalanobis distance to other points are illustrated with bigger and darker blue squares at the top of the graph. The median is the red dashed line (below), and the mean is the red dotted line (above). We believe that the points above 6 are suspiciously large, and that these airports could have been compromised by the foreign entities. These are airports with origin ID 10397, 12119, 12335, 13829, 13832, 13930, 14222, and 14512.

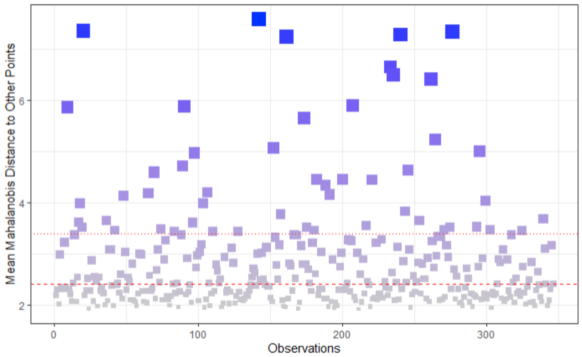


Figure 4a. Mahalanobis Mean Distance

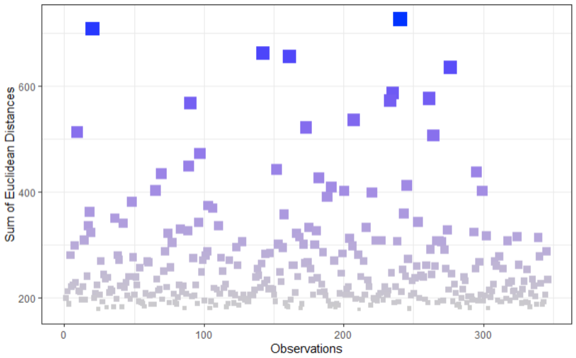


Figure 4b. Euclidean Distance

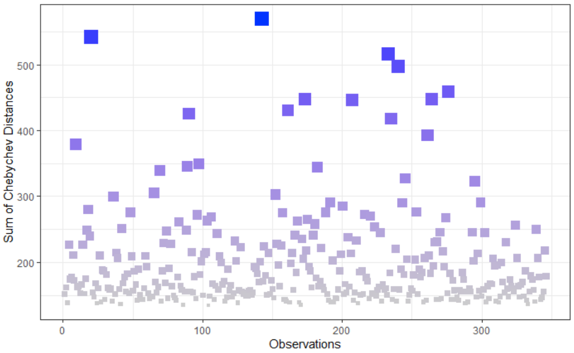


Figure 4c. Chebychev Distance

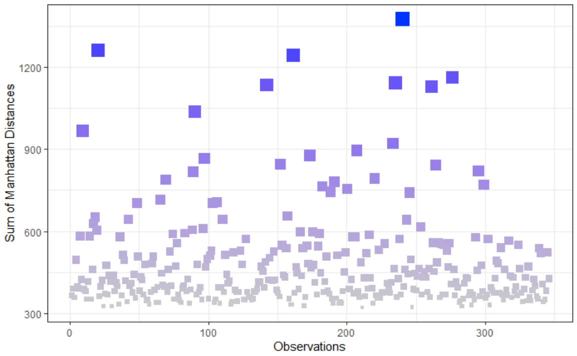
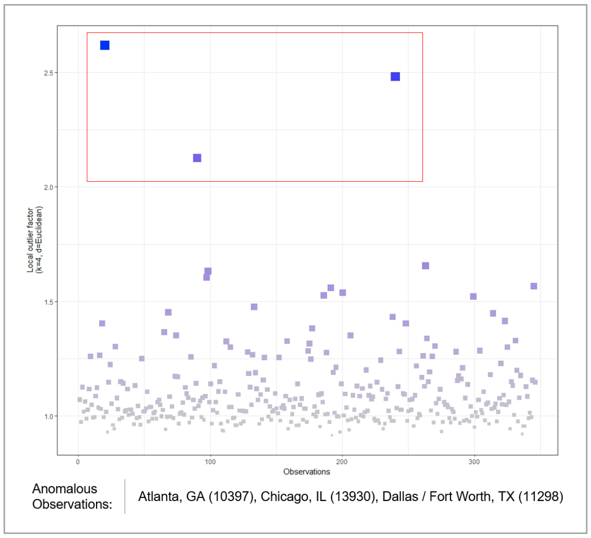


Figure 4d. Manhattan Distance

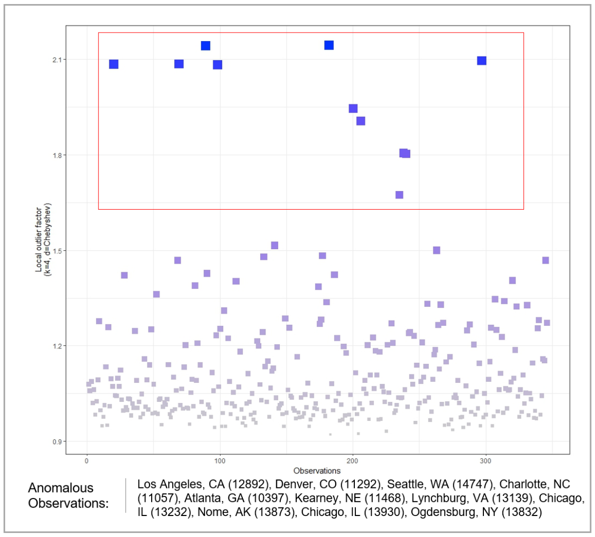
The Euclidean distance between **p** and **q** is the shortest path between them, the Chebychev distance is the maximum absolute difference, and the Manhattan distance is the sum of horizontal and vertical distances traveled between them. Figure 4b illustrates, for each observation, the sum of Euclidean distances to all other points. We identified the top 5 – 10397, 13930, 12119, 12335, and 14512 – as outlying. Similarly, we identified from Figure 4c, with sum of Chebychev distances top 4 (10397, 12119, 13829, 13930) as outlying and from Figure 4d, with sum of Manhattan distances top 7 (10397,12119 12335, 13832, 13930, 14222, 14512) as outlying. We notice some repetition here – for example, airport 10397 appears within all four distance measurements. This airport is possibly in great danger! This is further investigated later in the report.

Another method of anomaly detection we used is the Local Outlier Factor (LOF) approach. LOF is a density-based anomaly detection method, which involves k-nearest neighbors. Given a point, LOF measures the deviation from the k-nearest neighbors surrounding itself. We use this distance measure to arrive at a local density for each point. The points which we will determine to be outlying or anomalous based on this method will be those with lower densities, thus achieving a higher local outlier factor.



Anomalous Observations: Atlanta, GA (10397), Chicago, IL (13930), Dallas / Fort Worth, TX (11298)

Figure 5a. LOF Euclidean



Anomalous Observations: Los Angeles, CA (12892), Denver, CO (11292), Seattle, WA (14747), Charlotte, NC (11057), Atlanta, GA (10397), Kearney, NE (11468), Lynchburg, VA (13139), Chicago, IL (13232), Nome, AK (13873), Chicago, IL (13930), Ogdensburg, NY (13832)

Figure 5b. LOF Chebyshev

Seen above are the results of performing LOF on our PCA reduced data. Highlighted in red are the observations who stand out as local outliers based on the LOF algorithm. The airports which these points belong to are seen at the bottom of the graph. On the left, LOF using Euclidean distance results in three very distinct outliers among the observations, whereas using Chebyshev's distance as our distance metric results in a larger number of anomalous observations based on high LOF.

Clustering Techniques:

We also experimented with Two-Fold and K-means clustering techniques in an effort to cluster out the anomalous airports into a separate group from the un-invaded ones.

The Twofold model used is a two-step clustering method in which the algorithm first creates manageable subsets of data and then uses hierarchical clustering to merge subclusters into wider categories, determining the amount of clusters along the way. The algorithm's final clusters are provided below in figure 6. It picked out Airport ID {14771, 13930, 12953, 12892, 11298, 11292, 11057, 10397} to be likely invaded by foreign entities.

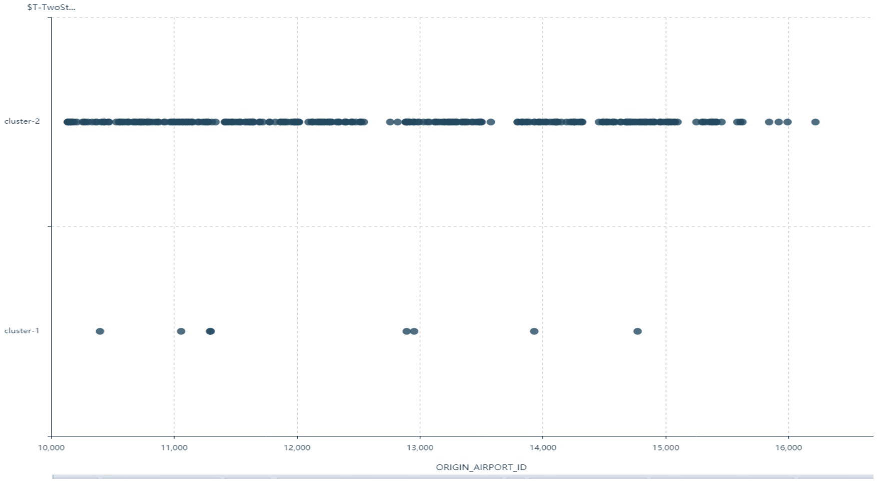


Figure 6. Two Fold Clustering

K means clustering randomly selects k points to be initial centroids and assigns each data point to its closest centroid cluster. The algorithm then re-calculates the new centroids of the clusters until the centroids don't move. This algorithm was tested using K = 2, 3, 5, and 10. K=5 yielded cluster 4 as the one with the most corresponding outliers to the other algorithms therefore it is the one depicted below in Figure 7 with airports 15582, 15008, 14716, 13983, 13832, 12917, 12223 and 12119 as the outliers.

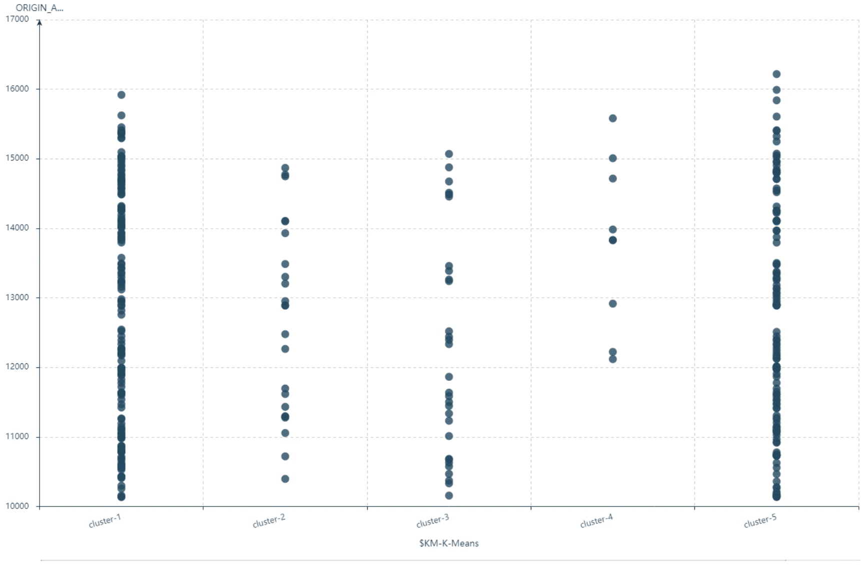


Figure 7. K-means Clustering with K=5 (cluster 4 being the outliers)

IBM Modeler *Anomaly Detection Clustering*

The data mining and visualization software IBM Modeler also features a model that dissects the data into K groups, then measures the *anomaly index* of each group and designates a point where this index is 2 or greater as an outlier. The *anomaly index* represents the ratio of the group deviation index to its average over the cluster that the case belongs to. The larger the value of the index, the more deviation the case has than the average. Points with an index greater than 2 are good candidates for anomaly because their deviation is twice the average. This algorithm was run with K = 3, 5 and 10 with K=3 yielding the most similar outliers to the other algorithms as shown in figure 8. The outliers were 10685, 10926, 111111, 11867, 12012, 12094, 12119, 12441, 13127, 13873, 14006, 14288, 14489, 14747, 14869, 15070.

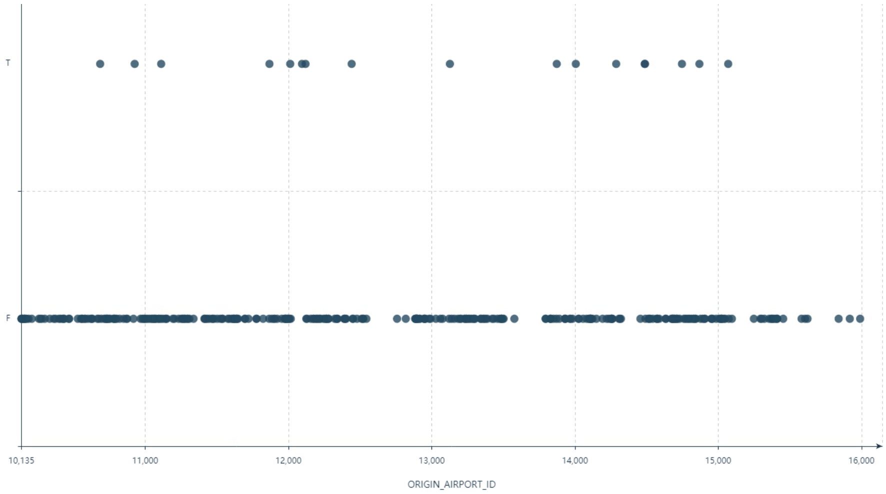


Figure 8. IBM Anomaly Detection Clustering with K=3

DBSCAN

DBSCAN is a density-based algorithm. The algorithm calculates the number of neighboring observations within a set distance for each observation, but is there any way to determine the exact value of this distance? Assuming that the minimum neighborhood size is set to 5, calculate the minimum distance at which an observation point can contain 5 neighbors and sort them from smallest to largest.

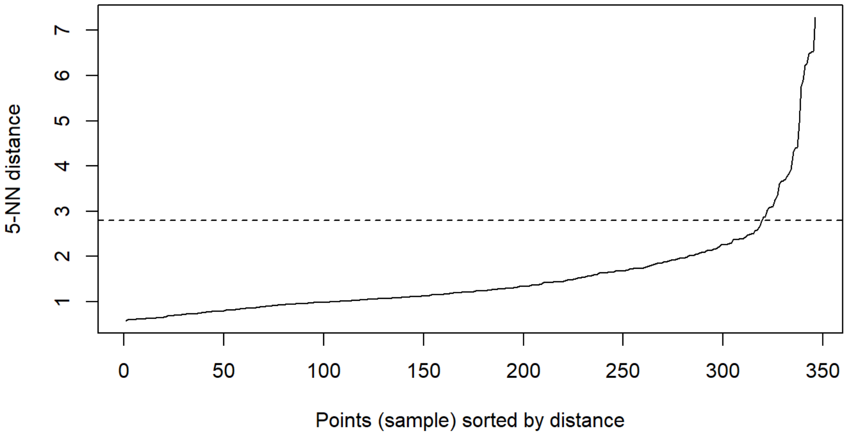


Figure 9. Threshold Selection

The dash line in the plot indicates an appropriate reachability distance. This is because observations from this value onwards require a larger reachability distance to satisfy the minimum of five neighbors, which also surfaces that they are in a sparse zone, far away from the vast majority of observations.

Here, we set the reachability distance to 2.8 and the minimum neighborhood size to 5. That is, an observation is taken at random and the Euclidean distance of 2.8 centered on it is computed. If there are five or more neighbors within that distance and there are no cores in those neighbors, then that observation is a core point. This is repeated until all observations are labeled. It is conceivable that those observations with less than 5 neighbors around them will be considered as outliers. With the above hyperparameter settings, we find 17 observations labeled as outliers. These possible outliers are 10165, 10397, 11013, 11447, 12119, 12223, 12335, 12519, 13241, 13388, 13829, 13832, 13983, 14222, 14254, 14512 and 14716.

As an example of validating our outlier airport record results, let's take a look at what's going wrong at airport 10397.

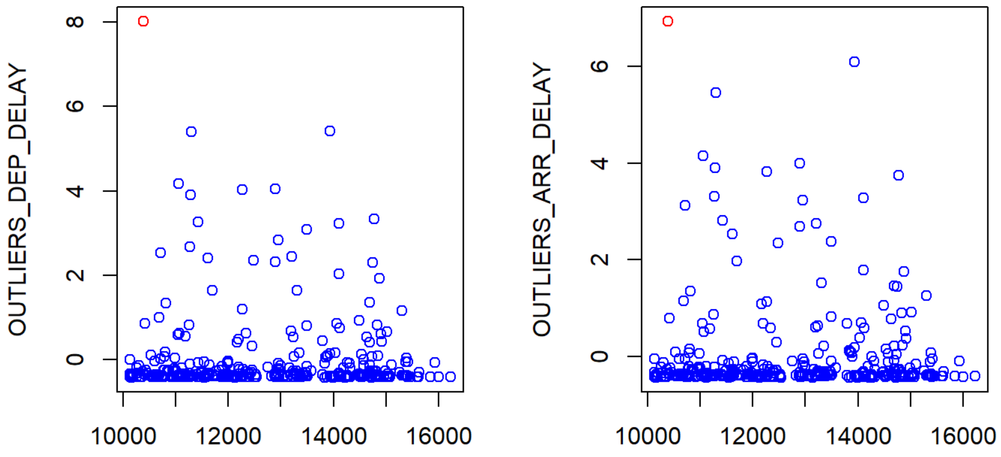


Figure 10. Airport 10397 Investigation

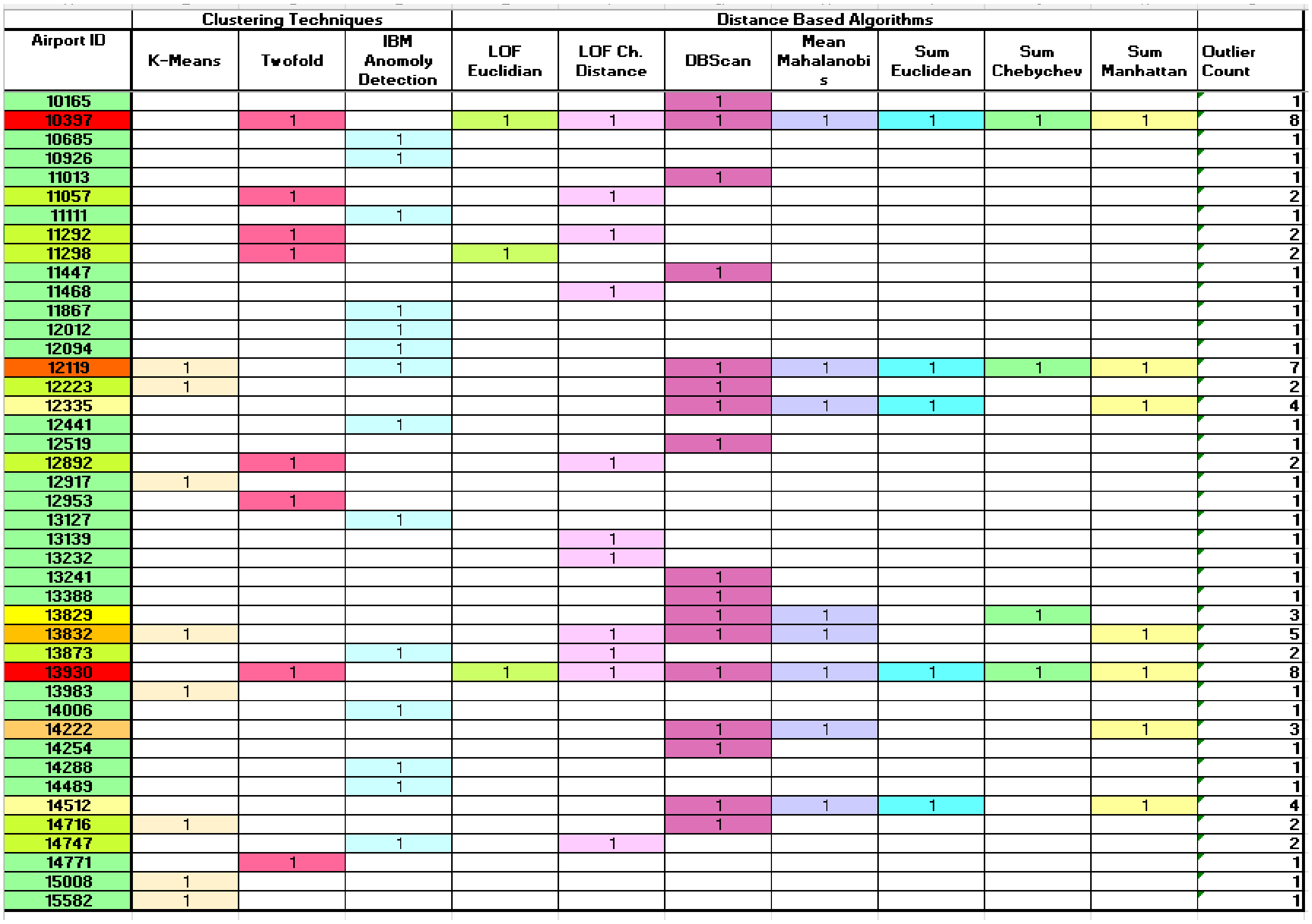
After investigating airport 10397, we found that the airport had irregular OUTLIERS_DEP_DELAY and COUNT_DEP_DELAY variables, indicating strange flight departure time and arrival time patterns thus confirming the algorithms' reasoning for why it is an outlier.

In conclusion:

According to our investigation and validation above, we highly suspect airports 10397 and 13930 (8 methods confirming anomaly) to be completely controlled by aliens, followed in likelihood by airports 12119 (7 methods indicating anomaly), 13832 (5 methods), 12335 and 14512 (4 methods) and 13829 and 14222 (3 methods). We caution the public not to book their air-travel from Atlanta, Chicago, Hagerstown, Ogdensburg, Iron Mountain, Rockford, Ogden and Pago-Pago until further investigation is complete.

As the dataset provided is not suitable for testing and all of the methods used are unsupervised, to validate our results, our options include investigating the individual flight patterns from the aforementioned airports or having our GhostBuster team deployed to do an on-site investigation.

Please see below a table summarizing our findings.



	Clustering Techniques			Distance Based Algorithms							
Airport ID	K-Means	Twofold	IBM Anomoly Detection	LOF Euclidian	LOF Ch. Distance	DBScan	Mean Mahalanobis	Sum Euclidean	Sum Chebychev	Sum Manhattan	Outlier Count
10165						1					1
10397		1		1	1	1	1	1	1	1	8
10685			1								1
10926			1								1
11013						1					1
11057		1			1						2
11111			1								1
11292		1			1						2
11298		1		1							2
11447						1					1
11468					1						1
11867			1								1
12012			1								1
12094			1								1
12119	1		1			1	1	1	1	1	7
12223	1					1					2
12335						1	1	1		1	4
12441			1								1
12519						1					1
12892		1			1						2
12917	1										1
12953		1									1
13127			1								1
13139					1						1
13232					1						1
13241						1					1
13388						1					1
13829						1	1		1		3
13832	1				1	1	1			1	5
13873			1		1						2
13930		1		1	1	1	1	1	1	1	8
13983	1										1
14006			1								1
14222						1	1			1	3
14254						1					1
14288			1								1
14489			1								1
14512						1	1	1		1	4
14716	1					1					2
14747			1		1						2
14771		1									1
15008	1										1
15582	1										1

Figure 11. Outlier Detection Method Comparison

Stay safe,

Alien Invasion Prevention Squad