AVIATION
ISSN: 1648-7788 / eISSN: 1822-4180
2022 Volume 26 Issue 4: 209–216
https://doi.org/10.3846/aviation.2022.18025

A METHOD FOR AUTOMATIC AIRPORT OPERATION COUNTS USING CROWD-SOURCED ADS-B DATA

Nicoletta FALA [1, *], Christos FALAS [2], Anastasios FALAS [3]

[1]*Mechanical and Aerospace Engineering, Oklahoma State University, 74078 Stillwater, USA*
[2]*Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, United Kingdom*
[3]*Independent researcher*

Received 9 February 2022; accepted 20 April 2022

Abstract. Airports are tasked with counting and reporting their operations at least yearly. The counts are used at the local and national level to schedule maintenance, for research, and to receive funds, making their accuracy important. Historically, methods for counting operations at non-towered airports have relied on additional equipment at the airport or statistical estimates. In this work, we introduce a method to use crowd-sourced Automatic Dependent Surveillance – Broadcast (ADS-B) data from the OpenSky network to automatically count airport operations and report it separated by takeoffs and landings. We use two airports as case studies – Tulsa International Airport (TUL) and Purdue University Airport (LAF) – and compare the estimated operation counts from the ADS-B data algorithm to numbers reported through the Federal Aviation Administration's (FAA) Air Traffic Activity Data System (ATADS).

Keywords: airport operation counts, ADS-B, OpenSky, non-towered airports, crowd-sourced data, airport count models.

Introduction

Airport operation counts, i.e. the number of takeoffs and landings at an airport, help determine government funding, airport needs, and potential system improvements, justify infrastructure such as nearby control towers and navigational aids, and are used in environmental, safety, and operational studies. Airport managers also use operation counts for forecasting and decision making, such as in runway closures and maintenance. Additionally, more particular counts that include the type of operation (scheduled, commercial, training, etc.) are helpful in determining and supporting the necessary services for each airport. In the United States, the Federal Aviation Administration (FAA) and the Department of Transportation (DOT) use operation counts to maintain a plan for developing public-use airports, the National Plan for Integrated Airport Systems (NPIAS) (Federal Aviation Administration, 2018). Historically, such operation counts have been counted or estimated and reported on the Airport Master Record (FAA Form 5010). For airports with control towers, recording the number of operations they observe is a relatively easy task that is considered to result in reliable numbers while the tower is in operation. Similarly, flights under Instrument Flight Rules (IFR) can be counted through the air traffic control system. However, during non-operating hours, and at non-towered airports, those numbers can only be estimated based on statistical and mathematical models, resulting in a lack of accurate records (Federal Aviation Administration, 2007). Airports and research have historically used various methods to estimate operations to various levels of accuracy, mostly based on observed relationships and ratios of known aircraft and/or operations to an estimated number of total operations (Muia & Johnson, 2015; Johnson & Gu, 2017; Muia, 2007).

Past work evaluated the accuracy of statistical methods as well as aircraft traffic technologies and found that the simple methods of estimating operations using ratios cannot be supported by test results and recommended using activity sampling to estimate operations (Muia & Johnson, 2015). Muia and Johnson (2015) also recommended that counting technology needs to adapt to consider factors such as airfield layout and expected fleet. The addition of ADS-B transponders to the majority of the US-registered fleet provides us with an opportunity to validate past models with potentially more reliable data. The work discussed in this paper aims to use ADS-B data to automatically estimate operations without the need for extra equipment or input from human operators such as airport managers

*Corresponding author. E-mail: *nfala@okstate.edu*

or controllers. While the methods discussed are applicable to all airports for which ADS-B data is available, we test our algorithms at U.S. airports because of the ADS-B Out mandate which took effect on January 1, 2020. This mandate dictated that any aircraft operating in airspace defined in 14 CFR § 91.225 is required to be equipped with an ADS-B Out system. Airports at locations without ADS-B mandates are not appropriate testing locations because aircraft (especially smaller and older aircraft) are unlikely to be equipped with ADS-B transponders, creating large gaps between the counts of methods that depend on the data and those that are more observational in nature.

In this paper, we discuss the literature on currently available counting techniques (Section 1) and introduce algorithms to query a crowd-sourced database of ASD-B information to count operations at user-selected airports (Section 2). In Section 3, we test our algorithms on data from two airports: Purdue University (LAF), a Class D airport in Indiana, and Tulsa International Airport (TUL), a Class C airport in Oklahoma. Section 4 compares counts using our method to FAA counts at LAF over a period of 3.5 years to examine changes that resulted from the 2020 ADS-B Out Mandate. The last sections discuss our conclusions and limitations and recommend future work respectively.

1. Current counting techniques

A National Association of State Aviation Officials (NASAO) survey conducted in late 2006 and distributed to fifty state aviation agencies, seven airports, and four metropolitan or regional planning organizations (MPOs) identified methods used for estimating aircraft operations at non-towered airports as well as uses of the information (Muia, 2007). The survey found that the most commonly used method for obtaining operation counts at non-towered airports is simply asking the airport manager or other airport personnel what they believe that value should be, leading to over-estimated counts. Using airport guest logs and fuel sales, on the other hand, will not track all traffic – local pilots are not likely to sign the guest log, transient pilots may not purchase fuel, and touch-and-goes are not accounted for with either method. Acoustical counters collect information without the need for a human, but normally only record departures. These counters use a microphone near the runway which picks up the sound of departing aircraft at full engine power. The recordings are then post-processed to obtain the number of departures, which when doubled theoretically provides the total number of operations (Ford & Shirack, 1985). Acoustical counters are appropriate for single-runway airports with consistently loud traffic, such as jet-powered, turboprop, or multi-engine piston aircraft. Long runways (as short as 3,000 ft) require multiple counters (Muia & Johnson, 2015). While expensive, acoustical counters are used most often in literature (Muia, 2007). Pneumatic counters consist of a rubber tube connected to a counter which uses changes in air pressure as aircraft are rolling over it to count the number of operations. Pneumatic counters are mostly used on taxiways to observe ground movement to and from taxiways, to avoid excessive wear at faster speeds on runways. As a result, they are not able to count touch and go landings (Ford & Shirack, 1985). Cameras and video image detection methods are appropriate for centralized airports, with locations that are unavoidable to inbound or outgoing traffic, such as access points for terminal and hangar areas. These methods do not account for touch and go landings and are a lot more expensive than other methods. Visual observations are labor-intensive and infeasible for year-round counts but result in reliable observations.

Because of the cost associated with mechanical counting methods, researchers and airport managers have resorted to mathematical and statistical models. Multiplying an estimate for the operations per based aircraft (OPBA) by the number of aircraft based at the airport of interest, results in an estimate for the total operations at that airport. Similarly, an estimated ratio of instrument flight plans to total operations (IFPTO) may be used since flights on an IFR-plan are counted through the system. Muia and Johnson (2015) attempted to find consistent OPBA and IFPTO numbers from information and counts at towered airports that can be applied to similar non-towered airports. Through their work, they identified that statistically extrapolating four seasonal two-week samples of activity into a year-long operations count was significantly more accurate than any of the ratios.

Recent research has leveraged low-cost data-collecting technology to count operations at non-towered airports with limited personnel (Mott, 2018; Mott & Bullock, 2018). The technology uses elementary Mode S and Mode C signals, which do not report aircraft position but do report aircraft barometric altitude. Mott and Bullock (2018) use the reported altitude and the amplitude of the signal to estimate aircraft position. The method provided counts within 2.2% of actual operations during the evaluation period, but still requires the airport to obtain and later potentially maintain additional equipment. The developed transponder data system has been used for various applications such as measuring aircraft fleet mix (Yang et al., 2021) and for aircraft performance models (Yang & Mott, 2021).

2. ADS-B counting method

The OpenSky Network[1] provides open access to real-time ADS-B and Mode S data through receivers supported by volunteers around the world. The receivers are connected to the Internet and all transmitted data is archived in OpenSky's database. The OpenSky Network provides real-time data to the public through a map visualization, and historical data to researchers who request access through an API and an Impala Shell (Schäfer et al., 2014).

[1] https://opensky-network.org

ADS-B messages are decoded and interpreted before being stored in the database as state vectors. Each state vector contains information on identification, position, and velocity. Identification consists of the transponder's ICAO address, and potentially the aircraft's call sign. Position is stored in terms of latitude, longitude, and altitude. Velocity is reported in terms of lateral rate and a vertical rate. OpenSky also stores metadata such as timestamps (Schäfer et al., 2014).

The developed algorithms take advantage of the database structure and information to convert the state vectors into flights. As shown in Figure 1, there are two parts to the algorithm: the first determines and runs the necessary queries from the OpenSky Network database to retrieve data; the second post-processes the retrieved data to identify takeoffs and landings and count the number of total operations. The user is able to change parameters that determine how the algorithm operates (the size of the search box, the length of the time periods, saving and counting methods, etc.) but is only required to input two variables: the airport and the time period of interest.

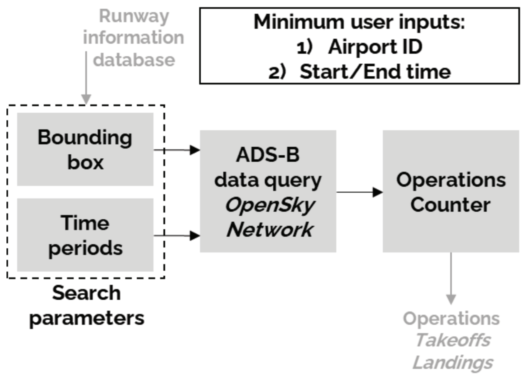

Figure 1. The developed algorithm requires the user to select an airport and a time frame, and outputs the number of takeoffs and landings at all runways of the selected airport during that time. The operations required to query the OpenSky Network (all functions before the *Operations Counter* function) are depicted in Figure 2. Pseudocode for the *Operations Counter* function is provided in Figure 4

The airport is determined by its three-letter identifier, and the time period from start and end local times. The user's airport ID input is used to retrieve information for the runways that make up that airport. A runway information database (United States Department of Transportation, 2020) provides the latitude and longitude of the end point of each runway, which we use to build a bounding box encircling all runways. The user's time input is first converted to Central Universal Time (UTC) to correspond to the OpenSky timezone, and then separated into one-hour periods to optimize the query. To improve the query performance, OpenSky clusters all data recorded in one-hour chunks, thus allowing distributed parallel processing of queries in an efficient manner. The pyopensky library creates an interface between the OpenSky Network database of historical data and our algorithms and allows us to download the requested state vectors by searching based on latitude/longitude and time bounds (Sun & Hoekstra, 2019; Sun et al., 2019).

ADS-B data is inherently inconsistent in its accuracy and highly dependent on the aircraft's sensors. The data retrieved from the query function includes missing datapoints and outliers. In this work, we used a moving average to remove some of the inaccurate behavior. We tried moving averages with varied memory (slower and faster

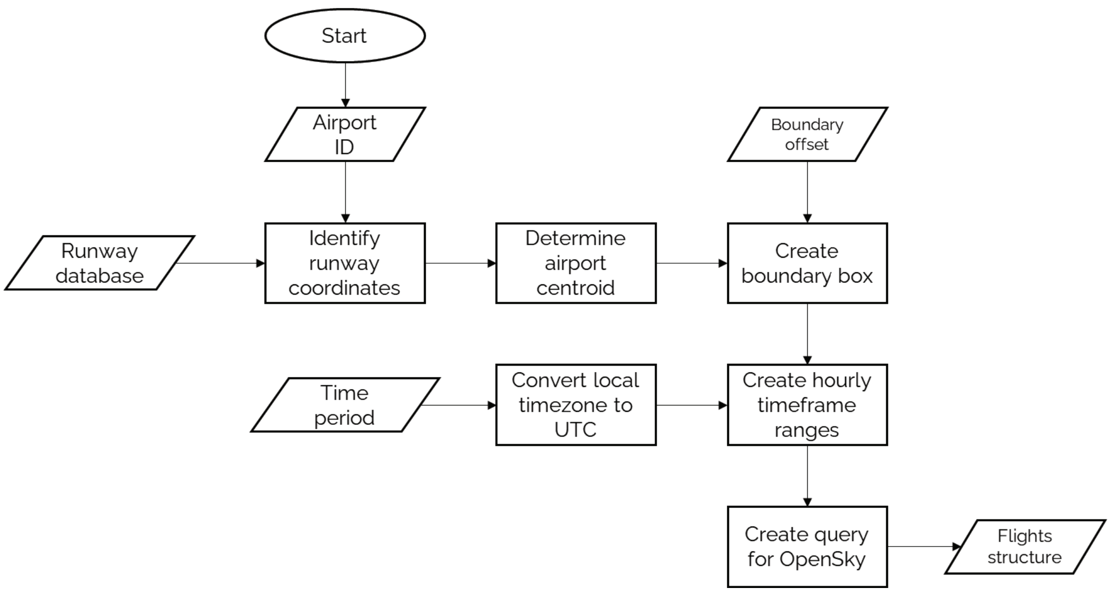

Figure 2. The process of automatically building and running a query for an airport relies on first constraining the data temporally and geographically. The algorithm takes the airport ID and a time period as user inputs to build a boundary box and generate and run a query. The query results in the *flights* structure, which contains all transmissions received during the specified time in the boundary box, and is used by the *Operations Counter* post-processing algorithm to count the number of takeoffs and landings

change) and arrived to an 8 second moving average as a tradeoff between losing data and having data that is not sufficiently smoothed. We then detect landings and takeoffs based on the aircraft's reported barometric altitude with respect to a pre-determined threshold.

There are three possible landing/takeoff combinations that the algorithm has to capture: (1) starting from the ground and taking off to depart or join the pattern, (2) starting airborne and descending to land, and (3) a touch and go. All three cases are indicated in Figure 3 with real-life examples from Tulsa International Airport (TUL) and Purdue University Airport (LAF). Both airports were chosen for having a good coverage by OpenSky. Figure 4 shows the algorithm in pseudocode for the takeoff/landing detection process and depicts the counter changes for the three aforementioned cases. Each time we observe two consecutive data points for the same aircraft with the altitude below and above the threshold respectively, the landing and takeoff counts both increase by one. Additional landings and takeoffs can only occur at the beginning or end of the data.

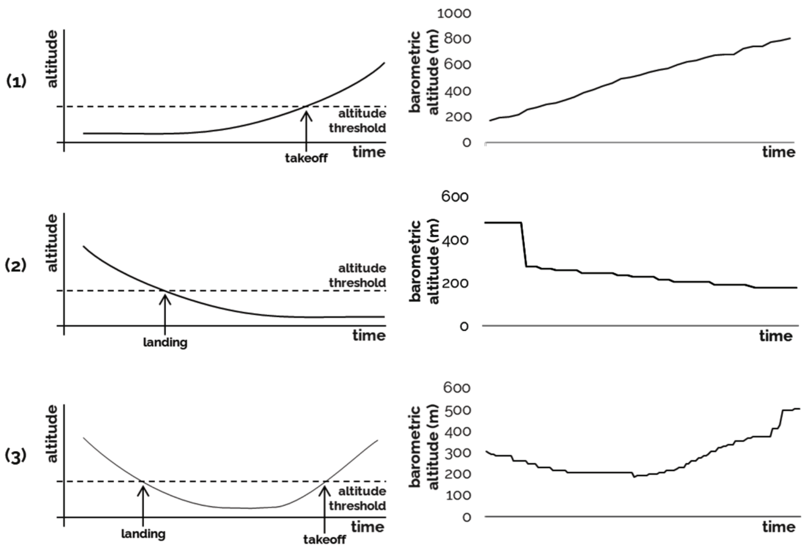

Figure 3. There are three operational scenarios that we need to account for in detection algorithms. The aircraft can (1) start its flight track on the ground and become airborne, (2) start airborne and land, or (3) takeoff and land in the same track, i.e. a touch and go. We define a flight track in the context of this algorithm as the period from the first time the aircraft is detected in the boundary box to the time it departed the boundary box or landed. The real flights (1) and (2) on the right are from TUL, and (3) is from LAF. These real flight examples also showcase some of the problems we encounter with the data with regards to jumps and resolution

```
FOR flight_j DO:

    Initiate variables
    landings = 0
    takeoffs = 0
    on_ground = False          landings = 0
                               takeoffs = 0
    IF y_1 < threshold DO:
        Flight is starting from ground
        landings = landings - 1
    END IF                     landings = -1
                               takeoffs = 0
    FOR y_i DO:
        Go through flight to identify threshold crossing
        IF y_i < threshold DO:
            on_ground = True
        ELSE DO:
            IF on_ground == True DO:
                landings = landings + 1
                takeoffs = takeoffs + 1
            END IF             landings = 0
            on_ground = False  takeoffs = 1
        END IF
    IF on_ground == True DO:
        landings = landings + 1
    END IF
    END FOR
                               landings = 0
END FOR                        takeoffs = 1
```

(1)

```
FOR flight_j DO:

    Initiate variables
    landings = 0
    takeoffs = 0
    on_ground = False          landings = 0
                               takeoffs = 0
    IF y_1 < threshold DO:
        Flight is starting from ground
        landings = landings - 1
    END IF                     landings = 0
                               takeoffs = 0
    FOR y_i DO:
        Go through flight to identify threshold crossing
        IF y_i < threshold DO:
            on_ground = True
        ELSE DO:
            IF on_ground == True DO:
                landings = landings + 1
                takeoffs = takeoffs + 1
            END IF             landings = 0
            on_ground = False  takeoffs = 0
        END IF
    IF on_ground == True DO:
        landings = landings + 1
    END IF
    END FOR
                               landings = 1
END FOR                        takeoffs = 0
```

(2)

```
FOR flight_j DO:

    Initiate variables
    landings = 0
    takeoffs = 0
    on_ground = False          landings = 0
                               takeoffs = 0
    IF y_1 < threshold DO:
        Flight is starting from ground
        landings = landings - 1
    END IF                     landings = 0
                               takeoffs = 0
    FOR y_i DO:
        Go through flight to identify threshold crossing
        IF y_i < threshold DO:
            on_ground = True
        ELSE DO:
            IF on_ground == True DO:
                landings = landings + 1
                takeoffs = takeoffs + 1
            END IF             landings = 1
            on_ground = False  takeoffs = 1
        END IF
    IF on_ground == True DO:
        landings = landings + 1
    END IF
    END FOR
                               landings = 1
END FOR                        takeoffs = 1
```

(3)

Figure 4. The pseudocode for the landing/takeoff detection indicates how the algorithm adjusts the landings and takeoffs counters accordingly for each of the three cases depicted in Figure 3. In Case (1), the algorithm will eventually detect a takeoff. However, since the flight starts on the ground, we start the landings counter at –1 to prevent the extra lading from the initial altitude being below the threshold. Case (2) is more straight-forward, with the aircraft being above the specified altitude threshold until landing. The algorithm ignores the if statement counting touch and goes, in black. In Case (3) the aircraft starts airborne but the flight track does not end with the landing, making that portion of the flight a touch and go

3. Testing and validation

Choosing airports for testing in the United States is challenging due to the multiple selection constraints. For comparison purposes, airports selected must have a control tower. Air traffic controllers report the number of operations daily to the FAA's Operations Network (OPSNET), which makes the data available in the Air Traffic Activity Data System (ATADS), providing us with a baseline against which to measure algorithm accuracy. Additionally, while the 2020 ADS-B mandate (Automatic Dependent Surveillance-Broadcast, 2019) has encouraged and incentivized most owners to equip their aircraft with transponders, they are not required in all airspace. Lastly, OpenSky relies on crowd-sourced data from receivers spread around the world, and reception in the United States is not as wide as Europe, further limiting the airports we can sample. For example, Aurora Municipal Airport (ARR) in Illinois has a full-time control tower and is within the Chicago Class B airspace, making ADS-B use mandatory, but does not have adequate coverage from OpenSky to be useful. While large airports like Chicago O'Hare (ORD) fulfill all requirements, the extensive number of operations makes historical data downloads impractical at the testing and validation stage of the research. We have selected Tulsa International Airport (TUL), a Class C airspace airport, for most of our testing because it fulfils all requirements and has a reasonable amount of traffic. We added Purdue University Airport (LAF), a Class D airspace airport, to some of the tests because it has a comparable amount of traffic to TUL but a different population of pilots and aircraft.

The ADS-B and Mode S messages report barometric and GPS altitude. However, both altitudes are reported with varying accuracy which is highly dependent on the aircraft's sensors. Figure 5 shows some of the inaccuracies observed from a sample flight at LAF. There is an offset between barometric and GPS altitudes, however, the magnitude of the offset is inconsistent. The errors are expected, as barometric altitude measures pressure differences and GPS (or geometric altitude) measures a distance. GPS will measure height with respect to either a geoid or ellipsoid Earth model, which can differ by up to 100 m. Some, but not all GPS receivers will add a geoid correction to output geometric height, creating discrepancies. The aviation industry uses barometric altitude as standard. Therefore, in this work, we used the barometric altitude for our algorithms and calculations. The observed data justifies using barometric altitude, with the elevation at LAF is 606 ft, or 185 m, and the lowest barometric and GPS altitudes shown in Figure 5 being 183 m and 145 m respectively.

Because the accuracy of the reported altitude is not consistent, detecting a landing and/or takeoff is highly dependent on the offset between the airport elevation and the threshold we choose for the algorithm. Table 1 makes the dependency clearer by reporting the number of landings and takeoffs detected during the month of December 2020 at TUL, where elevation is 220 m, using different thresholds. Table 2 reports the number of landings and takeoffs detected for one day segments, on December 1 and December 10, 2020. There is no preset threshold altitude that will result in minimizing the landings and takeoffs lost. To account for these changes, the algorithm iterates through an array of threshold altitudes (user-selected or default) and adds the maximum landings and maximum takeoffs detected to calculate total operations. In the case of Table 1, with the 50 ft threshold jump, the total operations would be calculated by adding 1,248 landings and 2,245 takeoffs, resulting in 3,493 total operations.

Table 2 presents information on the number of landings and takeoffs at TUL and LAF for two days in December of 2020 with varied parameters in the algorithm. As a result of changing the algorithm parameters, we make several observations.

1. Like in Table 1, we varied the crossing altitude at which a plane is considered to be landing or taking off. LAF reported more consistent results in that 250 m, or about 65 m above the airport's surface, always resulted in the maximum number of landings,

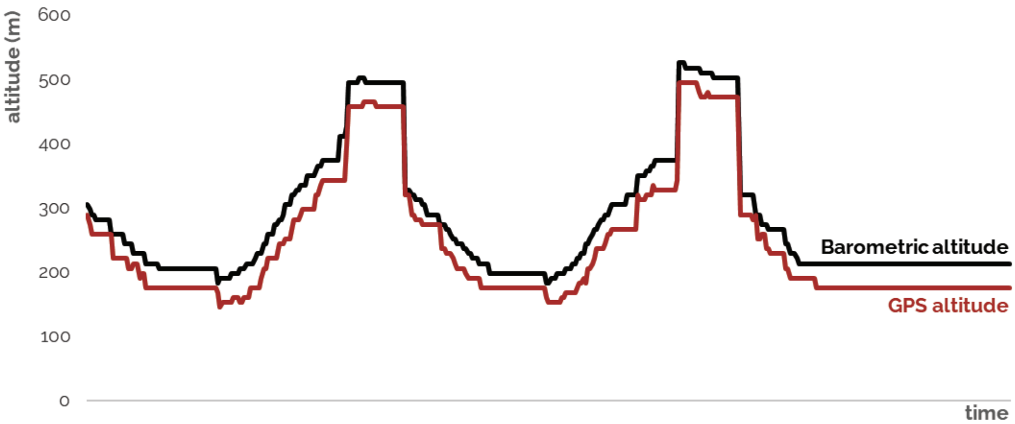

Figure 5. The two reported altitudes (barometric and GPS) both vary in their accuracy. In this case, the GPS altitude is lower than the barometric altitude throughout the flight, with the two altitudes occasionally converging

Table 1. The number of detected landings and takeoffs changes based on the selected threshold. The maximum landings and maximum takeoffs are not detected at the same threshold altitude. Note that the control tower at LAF does not operate overnight, but the algorithm continued to detect landings throughout the month

December 1, 2020–December 31, 2020			
TUL			
Threshold (m)	Landings	Takeoffs	Total Operations
150	559	573	1132
200	1248	1715	2963
250	1192	2123	3315
300	1108	2245	3353
350	952	2059	3011
FAA-reported	N/A	N/A	5500
LAF			
Threshold (m)	Landings	Takeoffs	Total Operations
150	1741	1742	3483
200	2595	2638	5233
250	2641	2704	5350
300	2502	2572	5074
350	2261	2327	4588
FAA-reported	N/A	N/A	6766

takeoffs, and total operations. TUL is inconsistent and the same threshold may not maximize both landings and takeoffs. Additionally, the same threshold did not maximize operations on the two days tested.

2. The "optimum searching" row has the algorithm iterating through the same range of threshold values (150–350 m) at 5 m intervals to identify the maximum operations we can possibly detect and compare the difference to the FAA-reported totals.
3. Our algorithms seem to be performing better at LAF than TUL, when compared to the FAA-reported totals. Additionally, the number of landings and takeoffs at TUL show large differences. While we do not expect the number of landings and takeoffs to be exactly the same, since some operations call for overnight stays, among other reasons, the large difference is a cause for concern. Our hypothesis for these inconsistencies is that the boundary box offset distance from the runway ends is to blame, so we changed the offset from 0.5 Nautical Miles (NM) to 2.0 NM to obtain the numbers in the parentheses. While ADS-B is expected to transmit data at a frequency of 1 Hz (Tabassum & Semke, 2018), that is not always the case in reality. An aircraft at a high speed could fly through the 0.5 NM distance and cross below the altitude where it would be detected without transmitting an updated signal, resulting in our algorithm missing the landing. Changing the offset made a bigger difference at TUL than LAF, which was expected due to

Table 2. The same threshold altitude analysis for a one-day range (December 1, 2020, and December 10, 2020) reveals that the "optimum" threshold altitude is not constant, especially at TUL. The maximum takeoffs occurred by setting the altitude threshold at 300 m on first test day, but 350 m on the second test day. In the "optimum searching" row, we allowed the algorithm to search for altitudes that resulted in maximum landings and takeoffs by iterating the post-processing algorithm in the 150 to 350 m altitude threshold range with 5 m intervals. The numbers reported in parenthesis use a 2.0 NM boundary box, whereas the rest of the numbers reported are for a 0.5 NM boundary box

December 1, 2020			
TUL			
Threshold (m)	Landings	Takeoffs	Total Operations
150	0 (0)	0 (0)	0 (0)
200	59 (63)	51 (52)	110 (115)
250	69 (117)	115 (116)	184 (233)
300	50 (121)	118 (124)	168 (245)
350	48 (110)	112 (125)	160 (235)
Optimum searching	87 (123)	120 (127)	207 (250)
FAA-reported	N/A	N/A	313
LAF			
Threshold (m)	Landings	Takeoffs	Total Operations
150	176 (175)	168 (168)	344 (343)
200	230 (229)	227 (227)	457 (456)
250	230 (233)	231 (232)	461 (465)
300	229 (232)	230 (231)	459 (463)
350	222 (236)	222 (235)	444 (471)
Optimum searching	230 (236)	231 (235)	461 (471)
FAA-reported	N/A	N/A	536
December 10, 2020			
TUL			
Threshold (m)	Landings	Takeoffs	Total Operations
150	0 (0)	0 (0)	0 (0)
200	2 (2)	2 (2)	4 (4)
250	55 (61)	64 (70)	119 (131)
300	51 (109)	94 (101)	145 (210)
350	26 (116)	97 (108)	123 (224)
Optimum searching	72 (116)	98 (108)	170 (224)
FAA-reported	N/A	N/A	303
LAF			
Threshold (m)	Landings	Takeoffs	Total Operations
150	37 (37)	40 (40)	77 (77)
200	195 (197)	199 (199)	394 (396)
250	200 (204)	204 (205)	404 (409)
300	198 (206)	203 (207)	401 (413)
350	189 (213)	191 (214)	380 (427)
Optimum searching	200 (213)	203 (214)	403 (427)
FAA-reported	N/A	N/A	606

the difference in operations among the two airports, with TUL being frequented primarily by airline traffic, and LAF accommodating mostly training operations. The total number of operations increased, but perhaps more importantly, the landing and takeoff numbers became more consistent. The change in the offset comes at the tradeoff of now potentially counting more false positives in the case of LAF, where the number of operations detected with a high altitude threshold increased, suggesting that the algorithm was detecting missed approaches and go-arounds, an expected operation at a training airport. The boundary box offset value should therefore be a function of the type of operations expected at the airport of interest, with faster traffic requiring a larger boundary box, and smaller boundary boxes being more appropriate for slower operations such as those at a training airport. Increasing the boundary box offset at TUL also resulted in more consistent "optimum thresholds" between takeoffs and landings.

4. Time and method comparison

To establish the accuracy of the approach described compared to other, more costly methods, we wanted to run our analysis for the same timeframe as the ones used in other work. However, data in other work is dated. The testing and evaluation process of Mott (2018), for example, uses data in the period from September 12, 2016 to March 10, 2017. While some aircraft owners and operators started equipping their fleet with ADS-B out technology before the mandate deadline of January 1, 2020, ADS-B data was sparse that long before the mandate. During that same 180-day period, we counted just 2422 operations at LAF (1209 landings and 1213 takeoffs) using crowd-sourced ADS-B data – Mott counted 51,577 using Mode S and Mode C signals (Mott, 2018). We are therefore not able to run analyses that compare our results to those of other novel methods until more recent work is published that makes use of newer data.

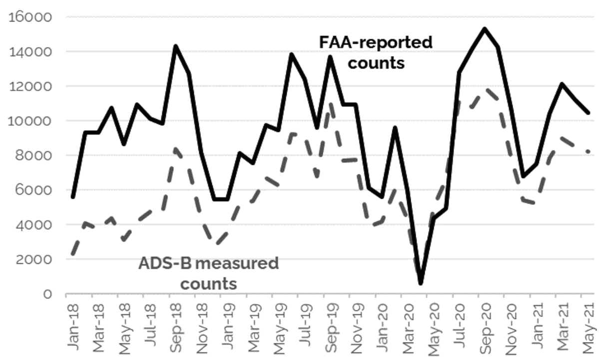

Figure 6. The gap between counts measured using crowd-sourced ASD-B and those reported by the FAA at LAF decreased starting approximately one year before the ADS-B mandate of 2020. The outliers in the number of operations around April of 2020 are due to COVID-19 shutdowns affecting the aviation industry

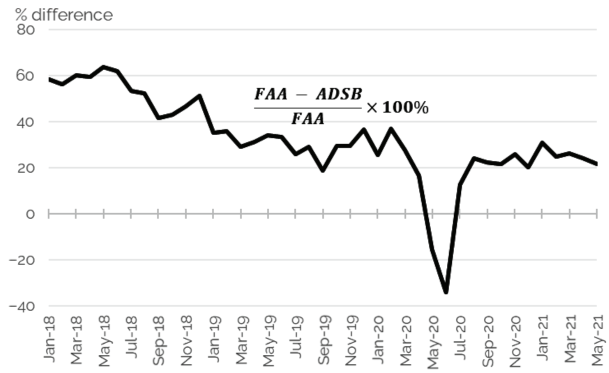

Figure 7. The difference between ADS-B measured and FAA-reported counts provides an insight into the proportion of the fleet equipped with ADS-B out. We observe a steady decrease in the difference which stabilizes at about 20%. The outliers in April and May 2020 correspond to small changes in Figure 6 but end up with a large absolute value due to the decrease in operations during the COVID-19 shutdowns

Instead, we ran an analysis to evaluate the "correctness" of automated ADS-B counts to the FAA ATADS information over time, in one-month increments. Figure 6 shows the monthly operation counts at Purdue University Airport from January 2018 to May 2021. The two lines correspond to the FAA ATADS counts and the counts detected and measured in crowd-sourced ADS-B data. Operations are cyclic, with more operations in the summer months and less in the winter. The FAA-reported counts are stable throughout the more than three years pictured – counts measured from ADS-B data show an increase over time, closing the gap between FAA and ADS-B numbers. Figure 7 makes the convergence clearer by reporting percept difference between FAA-reported numbers and ADS-B numbers and showing a steady decrease until the ADS-B mandate was in effect, converging to approximately 20% error. The outliers in both graphs in March, April, and May are a result of the COVID-19 pandemic. A lot of the traffic at LAF comes from student training, and since Purdue University terminated in-person classes on March 16, 2020 until the end of the semester the operation counts were greatly affected.

Conclusions

In this paper, we presented a method for using crowd-sourced ADS-B data near airports to report the number of landings and takeoffs. We developed an algorithm that takes user preferences to create a boundary box around the requested airport and count the total number of operations in the period that the user selects. This work can be used to report more accurate numbers to the FAA without the need to create additional infrastructure and maintain existing or new equipment. While using ADS-B data may not always result in accurate operation counts, depending on the location of the airport, the availability of data, and the type of operations (i.e. if not all aircraft frequenting the airport are ADS-B out compliant), using ADS-B data make new statistical methods possible. For example, while we have been basing the statistical models on the number of IFR operations or the number of aircraft based at the field, we can now use the ADS-B operations estimate as a starting point to build models that will in turn estimate the true total operations.

The biggest limitation to the method used is that we do not have a true count to compare to the ADS-B crowd-sourcing method. For example, while the tower at LAF counts operations during the day, it is not clear if or how the night-time operations are reported. An evaluation of the ATADS airport operation counts has revealed some inconsistencies that affect the work, with numbers reported being inflated. For example, some airports are adding over-flights to their itinerant count, not capturing true airport counts. Given the lack of transparency in operations counting and the inconsistencies, to ensure correctness, we need a period of visual observations, or flight data from a subset of aircraft in combination with the ADS-B data. Adding higher resolution to the OpenSky network, by having each airport feed data to the database through an antenna, would improve the reliability of counts.

The method presented has many advantages – it is very autonomous from an end-user perspective and the user does not incur extra costs in terms of equipment or personnel required. Additionally, it is possible to include more information with the counts and group operations accordingly. For example, we can separate operations based on the type of operation or the flight's transient/local status.

Future work will focus on improving the approach and expanding its capabilities. To improve the current algorithm, we can use visual observations at different airports so that we can tune our algorithm accordingly. Additionally, we plan to use commercial data from FlightAware and FlightRadar24 to evaluate whether they perform better given their more prevalent nature in the United States. Lastly, airports (particularly smaller airports) may benefit from an all-inclusive receiver which will feed data to the network while also counting operations at the airport.

Once more research is available in the post-2020 ADS-B mandate era, we will have an opportunity to compare the crowd-sourced ADS-B data approach to other methods, such as the one in Mott (2018), and compare the accuracy of results to evaluate whether more costly methods produce results that justify their cost.

Author contributions

NF was responsible for the design of the data analysis, providing oversight, and writing the paper. CF was responsible for the data collection and development of the algorithms and writing the description of the methodology. AF helped develop the data analysis and algorithms and reviewed the paper drafts.

Disclosure statement

We declare that we have no competing financial, professional, or personal interests from other parties.

References

Automatic Dependent Surveillance-Broadcast. (2019). *Out equipment and use. 14 C.F.R. § 91.225*. ADS-B.

Federal Aviation Administration. (2007). *An overview of terminal facility traffic counting: Attachment 1 Comparison of APO and CC traffic count FY 2007.* FAA.

Federal Aviation Administration. (2018). *National Plan of Integrated Airport Systems (NPIAS) 2019-2023.* U.S. Department of Transportation.

Ford, M., & Shirack, R. (1985). *Statistical sampling of aircraft operations at non-towered airports.* Federal Aviation Administration.

Johnson, M. E., & Gu, Y. (2017). Estimating airport operations at General Aviation airports using the FAA NPIAS airport categories. *International Journal of Aviation, Aeronautics, and Aerospace, 4*(1). https://doi.org/10.15394/ijaaa.2017.1151

Mott, J. (2018). Measurement of airport operations using a low-cost transponder data system. *Journal of Air Transportation, 26*(4), 147–156. https://doi.org/10.2514/1.D0117

Mott, J., & Bullock, D. M. (2018). Estimation of aircraft operations at airports using mode-C signal strength information. *IEEE Transactions on Intelligent Transportation Systems, 19*(3), 677–686. https://doi.org/10.1109/TITS.2017.2700764

Muia, M. J. (2007). *Counting aircraft operations at non-towered airports.* The National Academies Press.

Muia, M. J., & Johnson, M. E. (2015). *Evaluating methods for counting aircraft operations at non-towered airports.* The National Academies Press. https://doi.org/10.17226/22182

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., & Wilhelm, M. (2014). Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *Proceedings of the 13th IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN)* (pp. 83–94). IEEE. https://doi.org/10.1109/IPSN.2014.6846743

Sun, J., & Hoekstra, J. M. (2019). Integrating pyModeS and OpenSky historical database. In *Proceedings of the 7th OpenSky Workshop*, *67*, 63–72. https://doi.org/10.29007/mmsb

Sun, J., Vû, H., Ellerbroek, J., & Hoekstra, J. M. (2019). pyModeS: Decoding Mode-S surveillance data for open air transportation research. *IEEE Transactions on Intelligent Transportation Systems, 21*(7), 2777–2786. https://doi.org/10.1109/TITS.2019.2914770

Tabassum, A., & Semke, W. (2018). UAT ADS-B data anomalies and the effect of flight parameters on dropout occurrences. *Data, 3*(2), 19. https://doi.org/10.3390/data3020019

United States Department of Transportation. (2020, August 10). *Geospatial at Bureau of Transportation Statistics (BTS) Open Data Catalog: Runway Ends*. https://data-usdot.opendata.arcgis.com/

Yang, C., & Mott, J. H. (2021). Using transponder signals to model aircraft performance at non-towered airports. In *2021 IEEE Aerospace Conference (50100).* Big Sky, MT. https://doi.org/10.1109/AERO50100.2021.9438390

Yang, C., Mott, J., & Bullock, D. M. (2021). Leveraging aircraft transponder signals for measuring aircraft fleet mix at non-towered airports. *International Journal of Aviation, Aeronautics, and Aerospace, 8*(2), 1–18. https://doi.org/10.15394/ijaaa.2021.1563