Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)

Leveraging local ADS-B transmissions to assess the performance of air traffic at general aviation airports

Danae Z. Mitkas
David J. Lovell
Dept. of Civil and Environmental Engineering,
and Institute for Systems Research
University of Maryland
College Park, Maryland, USA
dmitkas@umd.edu, lovell@umd.edu

Sandeep Venkatesh
Seth B. Young
Dept. of Computer Science and Engineering,
and Center for Aviation Studies
The Ohio State University
Columbus, Ohio, USA
Venkatesh.64@osu.edu, young.1460@osu.edu

***Abstract*—This paper presents details of a novel hardware and software architecture for locally and automatically recording air traffic operations data at general aviation airports, including those that primarily serve smaller aircraft and may not have any other means, such as control towers, to collect such data. The platform is based on ADS-B data collected at 1090 and 978 MHz. We describe its deployment on the Amazon Web Services cloud computing environment. Various pre-processing and filtering stages are demonstrated to clean up the data. Some techniques for dealing with unreliable data are described. Finally, we show how to use the resulting data to compile air traffic performance metrics of interest for small airports, including aircraft approach speeds and runway occupancy times. These metrics may then be used to support enhanced models of airport capacity for these facilities.**

Keywords-general aviation, ADS-B, airport capacity

I. INTRODUCTION

A. Introduction

This paper describes a methodology for assessing the performance of air traffic operating within the vicinity of general aviation airports via the local collection and analysis of ADS-B data. Such airports typically lack automatic data collection capabilities, yet they would still like to be able to make an unbiased case for funding support for capacity improvement projects. Our methodology consists of collecting aircraft location and performance data at local airport environments from aircraft equipped with Automated Dependent Surveillance–Broadcast (ADS-B) technology, processing the data to determine and classify flights by aircraft and operational type using neural network learning models, and assessing elements of the operational performance of these aircraft to include aircraft speeds and runway occupancy times. These and other operational performance characteristics are known to be important contributors to the capacity of airports serving these operations.

B. Motivation

More than 13,000 public and private use airports in the United States accommodate the operations of fixed wing aircraft. While approximately 500 of these airports are designed to serve primarily commercial service aircraft, the remaining 12,500 serve almost exclusively smaller general aviation aircraft. More than 3,000 of these airports are supported by federal funding through the Federal Aviation Administration's (FAA) National Plan of Integrated Airport Systems (NPIAS) [1]. As such, understanding the facilities needs of these airports, and potential funding required to maintain appropriate levels of operational capacity at these facilities, is a critical infrastructure issue. Determining facility requirements, such as runway and taxiway infrastructure, depend on an understanding of the performance of aircraft operating these airports.

Compared to the comprehensive knowledge of operations at airports serving primarily larger commercial service aircraft, there is little current knowledge of aircraft performance in the vicinity of general aviation airports, particularly those airports serving small aircraft. As opposed to commercial service aircraft operations which operate in highly controlled and predictable environments, general aviation aircraft activity is much more variable in terms of aircraft performance and operations. For example, while commercial service aircraft follow strict arrival and approach procedures as dictated by the local air traffic control environment, a small aircraft at a general aviation airport may approach to land at an airport in a nearly ad-hoc manner, based solely on pilot judgement and visual perception. Small aircraft often perform various types of takeoffs or landings, or perform multiple "touch-and-go" operations for training purposes. In particular, the understanding of aircraft performance within the airport environment is critical to estimating an airport's operational capacity.

C. Background: Historical Airport Capacity Models

Guidance on estimating airport capacity analysis has been provided by the Federal Aviation Administration since the early 1980's with the publication of FAA Advisory Circular AC

150/5060- 5 *Airport Capacity and Delay* [2]. As of 2021, the FAA has yet to publish an update to this advisory circular, although portions of the document are under review for an update. The AC in its current published state describes methods for estimating the hourly operational capacity, annual service volume, and operational delays given various levels of demand, types of operations (ie. landings and takeoffs), runway configuration, certain meteorological conditions, and the aircraft fleet mix. Much of the modeling is focused on the use of nomographs and approximation charts, although some early design computer programs such as SIMMOD and ADSIM (both Federal Aviation Administration developed products) are described, as well. High level categorizations of inputs including aircraft types into four weight classes, two primary meteorological-based aircraft flight rules (VFR and IFR), and various runway configuration templates, provide coarse estimates of operational capacity. While technically applicable to all airports, the primary use of the advisory circular has been targeted to airports serving larger aircraft, typically in a commercial aviation environment. For airports serving primarily smaller aircraft, these models are far less robust [3].

One reason for the lack of robustness is lack of data for which to create richer models. Most airports that accommodate general aviation activity do not have air traffic control towers nor historically are equipped with other traffic surveillance technologies. As a result, it has been difficult to collect the data required to assess aircraft performance at these facilities.

D. Automated Dependent Surveillance – Broadcast (ADS-B)

With the recent widespread adoption of new ADS-B technology, the required data to understand aircraft performance around these airports is becoming widely available. This is supported by the FAA's mandate to require all U.S. registered aircraft be equipped with ADS-B technology when flying in controlled airspace, which includes many airport environments without air traffic control towers, as of January, 2020.

One benefit of ADS-B technology is the ability to capture raw ADS-B transmissions from aircraft in a local environment using inexpensive equipment. Using the methods described within this paper, the authors constructed and installed such equipment and designed an associated software architecture to collect and process the data to provide insights into air traffic performance on a variety of metrics. The assessment of these data is intended to be applied to enhanced airport capacity models that focus on these airports. Specifically, work performed through the U.S. National Academies' Airports Cooperative Research Program [4], developed spreadsheet-based models that may be adopted for smaller general aviation airport capacity estimation.

II. PREVIOUS RESEARCH

A. Airport capacity modeling

Previous attempts have been made regarding methods that would produce operation counts or the aircraft mix of an airport, data necessary for capacity estimation. A 2011 patent [5] claims to achieve low-cost aircraft detection in areas where ground surveillance radars do not exist or are limited. The system takes advantage of the acoustic emissions of the aircraft and translates them to "positional" and "aircraft type" information. A strong advantage, and at the same time disadvantage, of the system is that it does not require any additional equipment to be carried by the aircraft, since it relies solely on the acoustic emissions, meaning that any information collected is estimated and not communicated by the aircraft.

Newer research from Rashidi & Markovic (2020) proposes automated image-based aircraft tracking and record-keeping for airports [6]. The two main steps of this process include motion detection, which could provide operation counts, and aircraft recognition, to identify the aircraft mix. This method can reliably detect aircraft movements, especially for aircraft performing a single takeoff or a single landing. However, General Aviation airports often experience more complex aircraft activity, especially from training aircraft.

A mutual drawback of these two patents is that they can only provide information for the number of operations occurring at an airport and possibly the type of aircraft. Capacity estimation requires a combination of inputs to provide accurate results. Some of them are related to the airport characteristics, such as:

- Runway configuration,
- Control Tower Availability, and
- Runway Exits and Parallel Taxiway Availability.

Other required inputs are directly related to the airport activity and need to be carefully extracted and compute, to ensure trusted results. Some of these essential metrics are:

- Approach Speeds,
- Runway Occupancy Times (ROTs),
- Aircraft Separations,
- Touch-and-go operations, and
- Aircraft fleet mix.

Therefore, the use of ADS-B data was deemed the most appropriate method that would provide the necessary information. ADS-B messages transmit position, altitude, and speed data, as well as unique identification codes for each aircraft, which may be used to reference aircraft registry data to retrieve the aircraft's make, model, engine type, and other relevant information.

B. Use of ADS-B data

ADS-B data have been used in previous research for flight phase identification or for extraction of aircraft performance parameters. In 2016 Sun et al. approached the challenge of identifying and categorizing ADS-B data to create tools that would help handling these large amounts of data. The data get categorized in full or partial flight paths, based on the collected samples, with the use of clustering algorithms. Additionally,

Research sponsored by the Federal Aviation Administration Office of Airports, through the FAA Consortium for Aviation Operations Research (NEXTOR III).

because of the diversity of the data and the various traffic control procedures and aircraft characteristics, fuzzy logic was applied. Fuzzy logic allows the further clustering of the paths into flight phases which is an important part for understanding and modeling aircraft performance. Therefore, at the end of this two-step process the collected flight data are converted into meaningful clusters that will allow further research [7].

Later, the same team of researchers approached the modeling of aircraft performance parameters. Using ADS-B data from common commercial aircraft types, performance parameters were extracted from different flight phases. An aircraft performs differently in each phase and the different need to be clustered accordingly. The parameters studied include, but are not limited to, takeoff distance, average acceleration, vertical rate, cruise altitude, approach speed, rate of descent, and breaking distance. All of these provide an overall view of each aircraft's performance and create the basis for further modeling [8].

This research proposes a procedure for organizing the decoded ADS-B into flights, clustering the flights into phases, and extracting metrics useful for capacity estimation in small General Aviation airports.

III. ADS-B DATA COLLECTION

For this project, the research team has developed custom ADS-B data collection units based on the Raspberry Pi platform. This section describes the data collection hardware platform, the supporting software, and the database architecture.

A. Data Collection Apparatus

ADS-B messages are broadcast using pulse-position modulation (PPM) on a known, fixed frequency; hence, this is an open communication protocol that can be understood by any appropriately configured receiver. The most common configuration used in practice is a software-defined radio (SDR), which uses digital signal processing to complete all the necessary steps of tuning, demodulation, etc. Each of our units consists of the Raspberry Pi for processing, data storage, and communications, coupled with the SDR, a physical antenna, and a bandpass filter. To maximize the quantity of incoming data, we collect both 1090 Extended Squitter (1090ES) data at 1090 MHz and UAT data at 978 MHz. Since we are collecting both types of data, we install two nearly identical devices at each subject airport, one for each frequency. Externally, they look identical, but the SDR in each is tuned to its respective frequency, and the data formats for 1090ES and UAT are slightly different, so the internal processing is determined accordingly. A schematic of the broad data collection platform is shown in Figure 1.

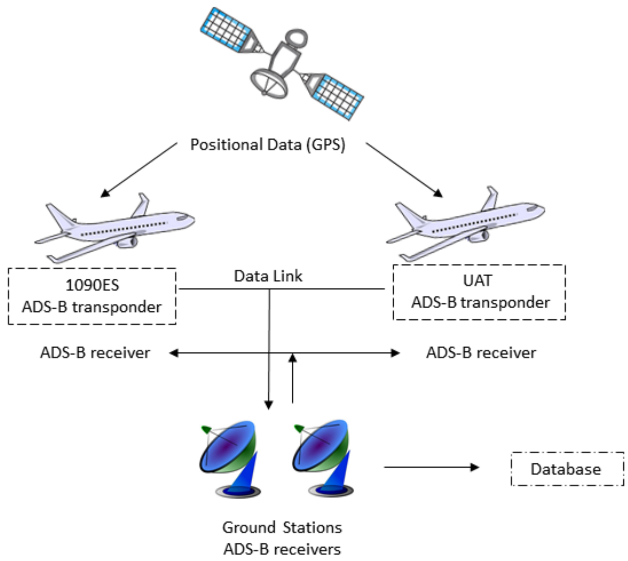

Figure 1: ADS-B system architecture

ADS-B receivers can collect aircraft data from any equipped aircraft that is detected within range. The gain on each radio is adjustable, and we have found that some trial and error is necessary to get the right value. When the gain is too high, too many messages are received, resulting both in extraneous data but also excessive corrupted messages due to message collision. Alternatively, when the gain is too low, aircraft of interest to the study, like those on final approach to the airport, cannot be detected at sufficient range.

The recorded ADS-B messages contain the following information:

- Aircraft Identification: 24-bit address assigned by the International Civil Aviation Organization (ICAO).
- Position: aircraft report their position twice per second, although not all messages are received. Aircraft position is provided in geodesic frame (WGS84). Position information includes latitude, longitude, Pressure Altitude and NUC, which indicates the integrity of the associated horizontal position data.
- Velocity: in east-west and north-south velocity and vertical rate.
- Some aircraft may broadcast status messages (e.g., emergencies, priority, capability, navigation accuracy category, operational modes etc.).

To ensure high quality ADS-B reception, the antenna must have good line of sight. When mounted inside a building, it is common for there to be blind spots in the reception coverage area. When this cannot be avoided, it is recommended to install in such a way that none of the arrival and departure paths are occluded by the blind spot.

B. Data Collection Software

The software system designed for the Raspberry Pi units allows them to run autonomously – once the units are running, no intervention is required by any personnel local to the airport. The units run on the Raspberry Pi OS operating system, which is a variant of Linux. The software is coded in a combination of BASH shell, C, C++, and Python programs. The software has the following features:

1. A configuration script that allows the user to select the airport, lat/long coordinates, altitude, and frequency of the receiver. The system automatically boots to these parameters and resumes normal operation in the event of a power interruption.
2. Data are decoded and stored on the local machine. Data are transmitted to an Amazon Web Services (AWS) portal in real time over the MQTT protocol, to support real-time mapping and flight display applications. Message failures in this protocol are tolerable, because the data are not archived and are only used for situational awareness.
3. Complete logs of system events, message transactions, etc., are recorded and stored locally.
4. Once an hour, any recent data and log files are uploaded via SCP to an AWS EC2 computing instance, and subsequently archived on the local computer, where they are retained for a month. In the event of a communications malfunction, this operation is re-attempted at every hourly upload event until it is successful. Files older than a month are deleted from local storage, because by this time they have been uploaded to AWS and stored in several places.
5. The system monitors real-time communications coming from AWS through the MQTT protocol, which allows researchers to ping the machines, remotely reboot them, tunnel into them to provide terminal window access, and change the gain levels on the radios.
6. A comprehensive upgrade engine has been implemented. During each hourly upload event, the units also check a database on the AWS server for system upgrades. If one is present, it is downloaded, installed, and the system is then rebooted. In this way, systematic upgrades to all working units can be initiated centrally, and the upgrades themselves occur automatically.

Figure 2 shows the overall system architecture. The devices are shown on the left, with their MQTT and DCP connections to the AWS Cloud. The rest of the project software is deployed at the AWS Cloud level.

The AWS Cloud software includes the following components:

1. An EC2 computing instance. This is where the data and logs files are initially uploaded. From here, the data files are decoded, filtered, and populated into a PostGreSQL database. The filtering step removes data too far from the subject airports to be informative for capacity estimation purposes. Some error-checking on altitude data also occurs here, and various flags are set to label the quality of the data. Individual records are consolidated into flights, and flight-level data are also sent to the Postgres database. Some summary statistics are gathered and sent to a Dynamo database, which drives a dashboard web page that users use to track system status.
2. Lambda functions.
   a. One processes the MQTT data submissions and populates a Dynamo DB with the last 5 minutes of real time data for the mapping and flight list web pages.
   b. One processes PULL requests from the mapping, flight list, and dashboard web pages, and invokes the AWS API to send responses to the associated HTTP requests.
3. S3 bucket. This is used as the final archive for all decoded message strings. It is also the project web server, hosting the mapping, flight list, and dashboard web pages, as well as a project information web page meant for research dissemination.
4. Cloudwatch. This process watches various system functions and issues alerts. This helps optimize the storage and processing levels, and warns project staff of possible remote unit malfunctions.

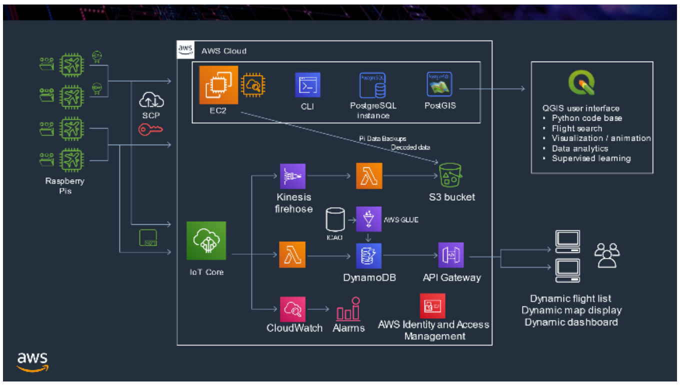

Figure 2: AWS system architecture

C. Data Collection Environments

The data collection units were deployed at three airports, the College Park Airport in College Park, Maryland (KCGS), The Ohio State University Airport in Columbus, Ohio (KOSU), and Republic Airport in Farmingdale, New York (KFRG). KCGS, illustrated in Figure 3, is a single runway airport serving approximately 30,000 general aviation operations annually. KOSU, illustrated in Figure 4, has three runways and serves approximately 100,000 operations annually. KFRG, illustrated in Figure 5, has two intersecting runways and serves approximately 200,000 operations annually. These airports, particularly KFRG and KOSU, serve high traffic volumes of small aircraft operations, including flight training, and hence made excellent locations for this research.

Figure 3: College Park Airport, MD (KCGS)

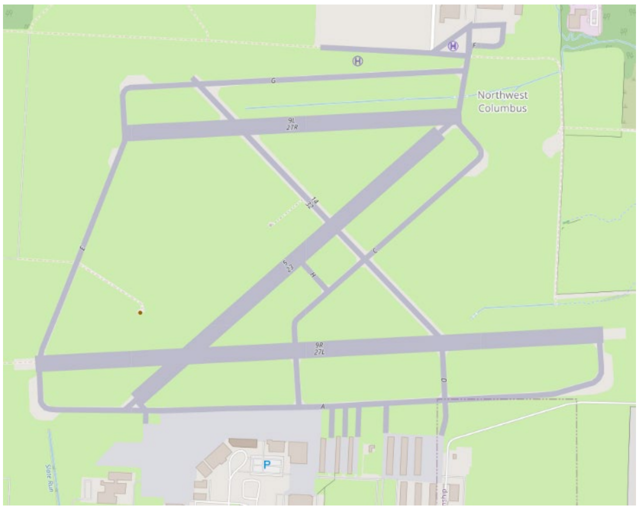

Figure 4: The Ohio State University Airport (KOSU)

The ADS-B receivers were placed in locations within administrative buildings at each airport, within line of sight to the airfield. At each of these locations, ADS-B data was captured from aircraft operating within a 7 nautical mile radius of the airport.

IV. DATA PROCESSING AND ANALYSIS

Over the initial 8-month period of system placement, approximately 30M individual ADS-B messages representing approximately 20,000 individual flights were captured. A subset of this data, specifically those data associated with flights with one or more operations at the airport, such as a takeoff or a landing, were entered into a "messages" table in the PostGreSQL database on the AWS EC2 instance for processing and analysis.

Figure 5: Republic Airport, NY (KFRG)

Data within each message relevant to assessing aircraft performance include the following:

- *DateTime*: Message Date Time stamp given in Unix epoch time.
- *ICAO*: Unique aircraft ICAO identifier of the aircraft transmitting the message.
- *Location*: Airport from which the ADS-B message was received.
- *Lat*: Latitude position from where the message was transmitted
- *Long*: Longitude position from where the message was transmitted
- *Alt*: Mean Sea Level altitude (ft)
- *GS*: Ground Speed (nm/hr)
- *TRK*: Tracking direction of the aircraft (in degrees)
- *ROC*: Rate of Climb (ft/min)

The messages are grouped into flights based on the unique aircraft ICAO identifier present in the messages, as well as a time window filter to assure that two separate flights of a given aircraft are not combined into one flight. Specifically, if there is a difference of time of more than 600 seconds in between two consecutive messages with a given ICAO, the message is considered to be a part of a separate subsequent flight by the aircraft. Each flight is recorded with a given unique Flight ID into a separate "flights" table within the PostGreSQL database. A JSON string of flight parameters from each message associated with the flight is added to each flight record. This string of information was then used to further investigate the performance of flights in the database.

A. Categorizing into operations using the LSTM Neural Network Modeling Framework

At a small airport, a given flight can perform various operations. A flight may have one operation, such as a takeoff or a landing, two operations, such as a takeoff, followed by a landing, or multiple operations, including multiple touch and go (landing followed by immediate subsequent takeoff) operations.

The process of categorizing flights into operations was a major focus of this study. The process began with creating various tools based on the QGIS geographic information system software platform to visualize and analyze the relevant performance characteristics of individual flights and manually classify the flights into operations. For example, Figure 6 illustrates a flight with a series of touch and go operations operating at KOSU. Figure 7 illustrates the changes in aircraft altitude, ground speed, and rate of climb, over the course of the flight. Figures 8 and 9 represent similar illustrations for an example aircraft landing at KOSU, Figures 10 and 11 represent similar illustrations for an example aircraft taking off at KOSU.

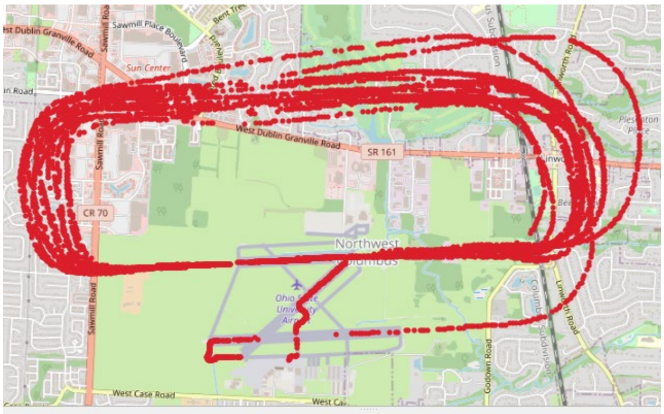

Figure 6: Sample flight path during touch and go operations

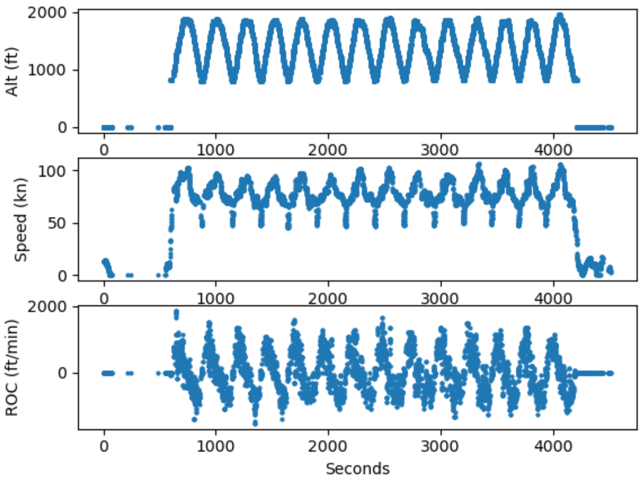

Figure 7: Touch and go parameters

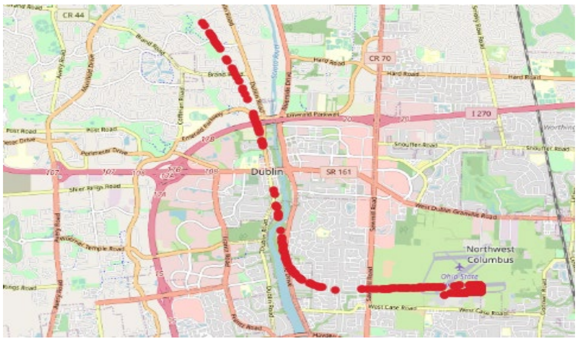

Figure 8: Sample flight path during landing at KOSU

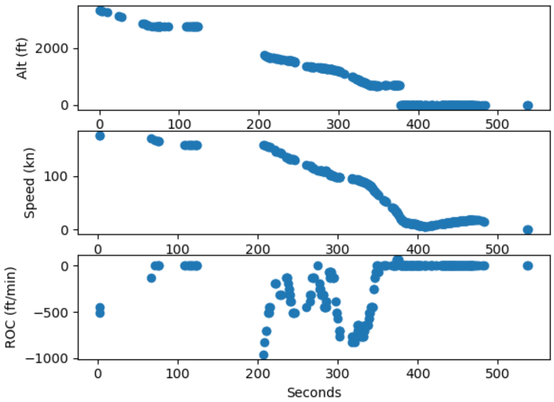

Figure 9: Landing parameters

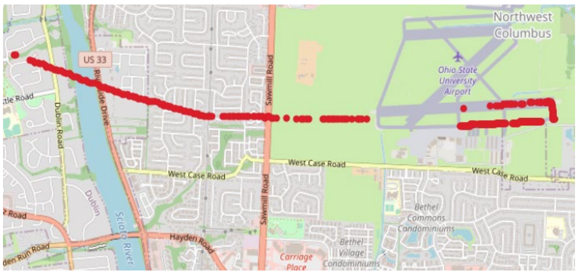

Figure 10: Sample takeoff at KOSU airport

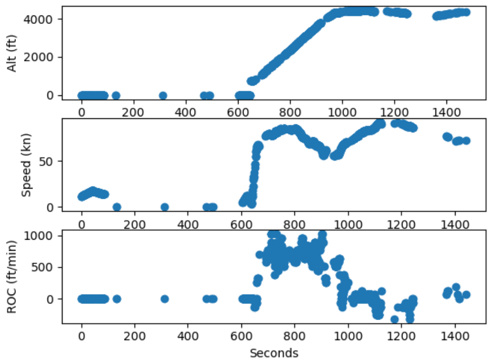

Figure 11: Takeoff parameters

Close inspection of Figures 5, 7, and 9 shows that there are some messages missing, creating possible gaps in a sequence of messages. This phenomenon occurs due ot the line of sight of the receiver or antenna pointing in another direction.

The clustered message of given flight is represented in a sequence of values of altitude, rate of climb, ground speed. Visual inspection of these values will help in classifying the data. Since visual inspection requires manual labor, a novel approach was developed in this research. This approach involved applying neural network to classify the flight sequences captured using the ADS-B receiver.

Long short-term memory (LSTM) [9] is a deep learning model based on recurrent neural networks, that is used for classification and prediction of time series data. The flight data captured from the ADS-B receivers can also be viewed as a sequence of values over a period. This attribute of the flight data made us pick the LSTM model to identify the sequences.

The dataset for training the LSTM network was manually classified by analyzing the graphs of parameters altitude, ground speed and rate climb. A total of 7651 flights were manually classified by visually analyzing the parameters. Table I shows the distribution of the labels across the dataset.

The LSTM model was used to train the model using TensorFlow. The model consisted of 4 hidden LSTM layers. The activation function softmax was used in the last layer since we were classifying 8 different types of operations. While training, categorical cross-entropy and rmsprop were used as the loss function and optimizer, respectively. Since the total number of messages can vary between flights, the model was trained with batch size set to 1 and adapted for a varying length of the sequences.

TABLE I. DATASET DISTRIBUTION BY OPERATION TYPE

Code	Operation Type	# Flights
0	No Operation	76
1	Touch and go	521
2	Low approach	1
3	Landing and then take off	55
4	Take off and then landing	202
5	Landing	2791
6	Take off	3150
7	Taxiing	855

TABLE II. CONFUSION MATRIX OF LSTM BASED OPERATION CLASSIFICATION OF FLIGHTS

		Predicted operation (Code)							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Actual Operation (Code)	**0**	0	0	0	0	0	0	0	3
	1	0	5	0	0	1	5	1	0
	2	0	0	0	0	0	0	0	0
	3	0	0	0	1	0	0	0	0
	4	0	0	0	0	1	0	0	0
	5	0	0	0	0	0	116	0	2
	6	0	0	0	1	0	0	117	3
	7	0	0	1	0	1	1	1	26

An accuracy level of 94% was achieved during the training stage, and the test accuracy was 92.9%. The test data had also been manually classified, but none of them were included in the training set. As shown in the resulting confusion matrix illustrated in Table II, reduced accuracies were primarily found for touch and go operations being misclassified as landings or takeoffs. Specifically, the model would fail to classify a flight with a single touch and go. Instead, it was classified as a landing or a takeoff, depending on the final stage of the flight. It should be possible to increase the accuracy by identifying specific canonical operations in the data set and ensuring that they are included that they included in the training dataset.

B. Handling unreliable altitude data

During our analysis of the flights, it was observed that some of the aircraft were broadcasting incorrect altitude readings, or at least that these data were being received or demodulated incorrectly. The flight parameters shown in Figure 12 demonstrate such a case. After analyzing the speed and rate of climb parameters it can be concluded that the flight is taking off. However, the altitude data is not accurate and would be detrimental if downstream classification efforts were made without rectifying this situation.

After analyzing the various flight data, we concluded that a good test for unreliable altitude data is to check for flights with calculated absolute rates of climb greater than 7500 ft/ minute. In these cases, we replace the actual (flawed) altitude data with estimated data constructed by numerical integration of the rate of climb data. Figure 13 shows the result of this process for the same flight as before, and it is clear that the aircraft is taking off.

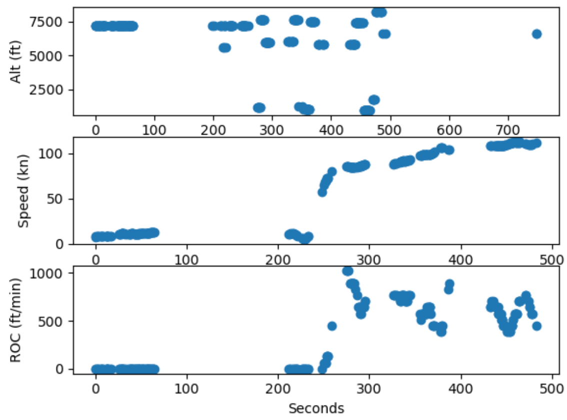

Figure 12: Invalid altitude reading.

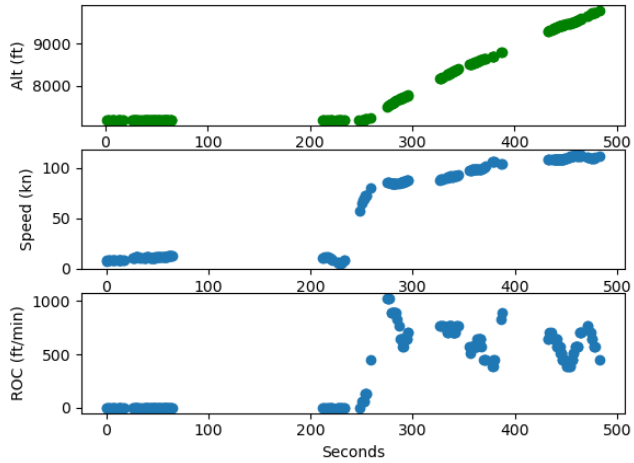

Figure 13: Integrated alitude using rate of climb data

C. *Measuring performance*

Having identified the different aircraft operations, the analysis of the data focused on estimating flight performance metrics to support capacity estimation. Approach Speed and Runway Occupancy Time (ROT) were the first metrics to be investigated, being two of the most important to capacity modeling. The results shown are for the KOSU (Ohio State University) and the KFRG (Republic Airport at Farmingdale) airport. Both are General Aviation airports with significant activity, and at the same time with different type of operations. KOSU has considerable training activity, with aircraft performing continuous touch-and-go's and an often-changing pattern, while KFRG accommodates both training aircraft and jet services and maintains a rather steady pattern, with aircraft landing mostly on runway 32.

1) Approach Speed: Approach speed is the ground speed that aircraft have before approaching the runway to land (i.e. during their final approach) and varies based on the aircraft type. To measure the approach speed all messages from aircraft heading to the runway (based on their track), and with a decreasing altitude (and negative rate of climb) are captured and averaged by runway and aircraft type. The overal results for each runway used for landing and for the different aircraft types operating at these airports, are summarized in Table III.

TABLE III. AVERAGE APPROACH SPEED

Airport	***Runway used for landing***	***Avg. Approach Speed per aircraft type (knots)***	
		Fixed wing single engine	***Fixed wing multi engine***
KOSU	9L	66	62.5
	9R	71	68
	27R	70	69.5
	27L	71	73
	5	58	63
	23	59	60
KFRG	14	62	82
	32	68	83

The results are slightly higher for the longer runway at KOSU (9R/27L) since this runway serves larger aircraft which tend to approach at a higher speed. This is also the case at KFRG, where multi-engine aircraft approach with a significantly higher speed. Lower approach speeds are found on runways accommodating solely the smaller single engine aircraft (KOSU 9L) and aircraft operating on crosswind runways when headwinds tend to be greater (KOSU 5/23).

2) Runway Occupancy Time (ROT): ROT is defined as the time that an aircraft is occupying the runway. For arriving aircaft this time starts from the moment an aircraft crosses the runway threshold upon landing to the time it exits the runway.

ROT is an increased limitation to runway capacity and holds an important role in an airport's capacity estimation. Therefore, collecting accurate data and carefully minimizing ROT will increase the runway's capacity. It has been observed that ROT is largely affected by the exit used, the aircraft type, the approach speed, and the distance from the following approaching aircraft[10][11]. The results collected for ROT, based on airport, runway and exit used and divided by aircraft type, are collected in Table III. Blank spaces indicate that either the runway or the exit is not used by that aircraft type. The blank cells of the table indicate that either the runway is not frequently used by that aircraft type (multi engine aircraft do not operate on short runways), or the exit cannot be used by that aircraft type. Specifically, multi engine aircraft will not use the exit closer to the runway threshold and single engine aircraft rarely taxi to the end of the runway if there is a sooner exit. Figures 14 and 15 indicate the different runways and different exit locations at the two airports.

TABLE IV. RUNWAY OCCUPANCY TIME (ARRIVALS)

Airport	*Runway used*	*Exit Used*	*Runway Occupancy Time (sec)*	
			Fixed wing single engine	*Fixed wing multi engine*
KOSU	9L	End of runway	43	--
	9R	C	34	--
	9R	D	48	34
	9R	End of runway	--	64
	27R	End of runway	49	67
	27L	D	28	--
	27L	C	45	30
	27L	End of runway	--	98
	5	End of runway	82	--
	23	End of runway	80	--
KFRG	14	B	29	--
	14	G	32	36
	14	A5	--	50
	32	A5	29	30
	32	A4	33	34
	32	B	42	40

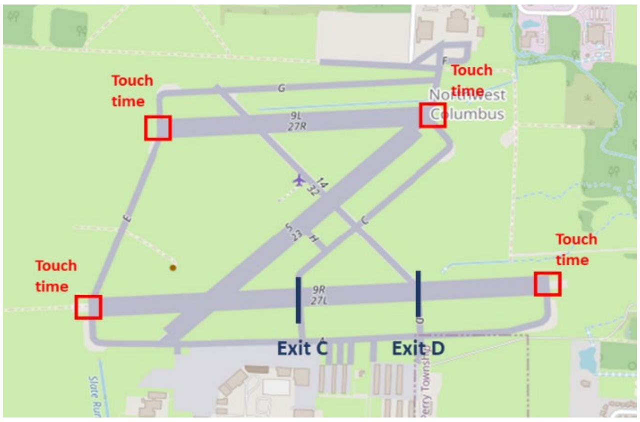

Figure 14: KOSU runway exit and taxiway locations

The results for the ROT were mostly anticipated. The further the exit used from the threshold, the more time required on the runway. However, while expected to have lower values in the case of multi-engine aircraft, since as seen previously they approach with a higher speed and therefore would cover the distance in less time, it was not confirmed. After examining the case of multi-engine aircraft closer, it was observed that even though the aircraft touch the runway at a higher speed, in many cases they slow down more abruptly and then taxi at a low speed until the exit.

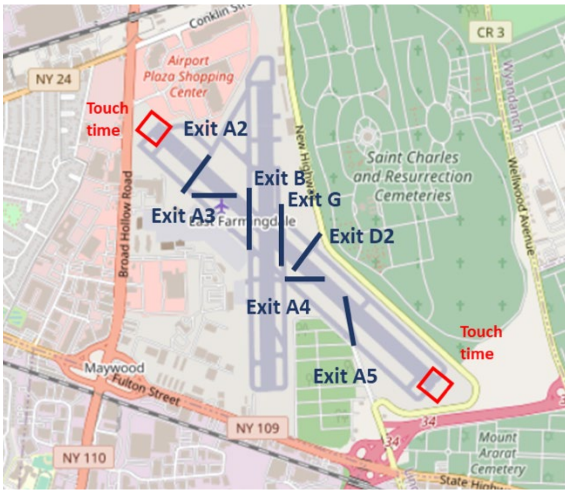

Figure 15: KFRG runway exit and taxiway locations

V. FINDINGS

Overall findings from this research to date reveal that leveraging ADS-B data to further understand the performance of general aviation aircraft operating within the airport environment is a feasible and productive exercise. Data collected from equipment built at relatively low expense and stored onto cloud-based database and computing infrastructure provide the ability to collect and store rich volumes of data from several locations quite efficiently. Furthermore, the data collected provide highly accurate information for all phases of flight, including aircraft surface movements. As third-party sources of ADS-B data generally are not available in raw form down to the airport surface, the build out of the project's exclusive equipment was found to be an essential part of the data collection process.

In addition, initial performance measures of aircraft approach speeds and runway occupancy times appear to be highly valid.

VI. FUTURE RESEARCH

From these initial findings the research is motivated to deepen this analysis by investigating performance characteristics based on each of the specific runway operating environments, for specific aircraft types, and by specific meteorological conditions such as wind direction and speed, and cloud ceiling and visibility conditions.

In addition, to study various features of a small airport, the flight speeds during landing and takeoff will aide in understanding airport capacity. Some preliminary work has been done on dividing the flights into various phases using the density-based clustering algorithm DBSCAN. DBSCAN clustered various messages according to their altitude, ground speed, rate of climb and track value. Even though this technique clustered the various phases of flight as shown in Figure 16, more work is required for it to reliably classify the phases of flights for a wide variety of flight patterns. In our future work we will develop technique to recognize various phases of flight [12].

Figure 16. Clustering phases of landing operation

This work will continue to further understand the performance characteristics of aircraft operating within these study airports. In addition, the study will deploy data collection equipment to other airports to gain further insight on how various runway configurations and other characteristics of the airport environment may impact aircraft performance. The ultimate findings from this work are planned to be used to further develop models for estimating the operating capacity of airports accommodating small general aviation aircraft.

ACKNOWLEDGMENT

This work has been sponsored under the by the Federal Aviation Administration's Office of Airports via the Administration's university consortium for Aviation Operations Research (NEXTOR III). This work is part of the FAA's larger effort to enhance operational capacity models for airports serving primarily smaller aircraft. The authors would like to acknowledge Mr. Kent Duffy, National Resource Expert for Airport / Airspace Capacity, FAA for his support of this work.

REFERENCES

[1] Federal Avsation Administation, *National Plan of Integrated Airport Systems (NPIAS), 2021-2025*, Published by the Secretary of Transportation Pursuant to Title 49 U.S. Code, Section 47103 on September 30, 2020.

[2] Fereral Aviation Administration, Advisory Circular 150/5060-5 *Airport Capacity and Delay*, September 23, 1983, Change 2 ver. December 1, 1995.

[3] Robert Horonjeff; Francis X. McKelvey; William J. Sproule; Seth B. Young. Planning and Design of Airports, Fifth Edition (McGraw-Hill Education: New York, Chicago, San Francisco, Athens, London, Madrid, Mexico City, Milan, New Delhi, Singapore, Sydney, Toronto, 2010).

[4] National Academies of Sciences, Engineering, and Medicine 2012. ACRP Report 79: *Evaluating Airfield Capacity*. Washington, DC: The National Academies Press. https://doi.org/10.17226/22674.

[5] Lee J.S., Nielsen R.O., Davis H.H. and Dunholter P.H., 2011, "Acoustic Airport Surveillance System", Patent No. US 8,059,489 B1, United States Patent.

[6] Rashidi A. and Markovic N., 2020, "Automated Image-Based Aircraft Tracking and Record-Keeping for Utah Airports", University of Utah.

[7] Sun, J., Ellerbroek, J., & Hoekstra, J., 2016, "Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods", In D. Lovell & H. Fricke (Eds.), 7th International Conference on Research in Air Transportation, Philadelphia, USA.

[8] Sun, J., Ellerbroek, J., & Hoekstra, J., 2017, "Modeling Aircraft Performance Parameters with Open ADS-B Data", 12th Air Traffic Management Research and Development Seminar (ATM2017), USA.

[9] Hochreiter, S. & Schmidhuber, J"urgen, 1997. Long short-term memory. *Neural computation*, 9(8), pp.1735–1780.

[10] Meijers, N.P. & Hansman R.J., 2019, "A Data-Driven Approach to Runway Occupancy Time", AIAA Aviation 2019 Forum, Dallas, Texas, USA.

[11] Kumar V., Sherry L. & Kicinger R.., 2009, "Runway Occupancy Time Extraction and Analysis using Surface Track Data", George Mason University, USA

[12] Sun, J., Ellerbroek, J., & Hoekstra, J., 2017, "Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance-Broadcast Datasets", Journal of Aerospace Computing, Information and Communication, DOI: 10.2514/1.I010520, the Netherlands

AUTHORS' BIOGRAPHIES

Danae Mitkas is a graduate research associate at the University of Maryland, graduating with her MS in Civil Engineering in May 2021.

David J. Lovell is Professor of Civil and Environmental Engineering at the University of Maryland, College Park. He holds a joint appointment with the Institute for Systems Research, and he is the Director of the Gemstone Honors Program. He is also Director of the NEXTOR III Aviation Operations Research Consortium led by the University of Maryland.

Sandeep Venkatesh is a graduate research associate at The Ohio State University, graduating with his MS in Computer Science in May 2021.

Seth B. Young is the McConnell Chair of Aviation at the Center for Aviation Studies, and Associate Professor of Civil, Environmental and Geodetic Engineering at The Ohio State University. Dr. Young is a co-Director of the NEXTOR III Aviation Operations Research Consortium.