

Air Force Institute of Technology

AFIT Scholar

Faculty Publications

2-2023

ADS-B Classification using Multivariate Long Short-term Memory–fully Convolutional Networks and Data Reduction Techniques

Sarah Bolton *

Air Force Institute of Technology

Richard Dill

Air Force Institute of Technology

Michael R. Grimaila

Air Force Institute of Technology

Douglas Hodson

Air Force Institute of Technology

Follow this and additional works at: <https://scholar.afit.edu/facpub>



Part of the [Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Bolton, S., Dill, R., Grimaila, M.R. et al. ADS-B classification using multivariate long short-term memory–fully convolutional networks and data reduction techniques. *J Supercomput* 79, 2281–2307 (2023). <https://doi.org/10.1007/s11227-022-04737-4>

This Article is brought to you for free and open access by AFIT Scholar. It has been accepted for inclusion in Faculty Publications by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.



ADS-B classification using multivariate long short-term memory–fully convolutional networks and data reduction techniques

Sarah Bolton¹ · Richard Dill¹ · Michael R. Grimaila² · Douglas Hodson¹

Accepted: 16 July 2022 / Published online: 12 August 2022

This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2022

Abstract

Researchers typically increase training data to improve neural net predictive capabilities, but this method is infeasible when data or compute resources are limited. This paper extends previous research that used long short-term memory–fully convolutional networks to identify aircraft engine types from publicly available automatic dependent surveillance-broadcast (ADS-B) data. This research designs two experiments that vary the amount of training data samples and input features to determine the impact on the predictive power of the ADS-B classification model. The first experiment varies the number of training data observations from a limited feature set and results in 83.9% accuracy (within 10% of previous efforts with only 25% of the data). The findings show that feature selection and data quality lead to higher classification accuracy than data quantity. The second experiment accepted all ADS-B feature combinations and determined that airspeed, barometric pressure, and vertical speed had the most impact on aircraft engine type prediction.

Keywords Multivariate long short-term memory–fully convolutional network · Automatic dependent surveillance-broadcast · Publicly available information · Open-source data · Classification · Machine learning

1 Introduction

Over the last three decades, storage on the internet increased by over 40,000% from 15.8 exabytes in 1993 to 6.8 zettabytes in 2020 [1]. While it is difficult to determine the exact number, as of February 2022, the size of the internet is

Richard Dill, Michael R. Grimaila and Douglas Hodson these authors have contributed equally to this work.

✉ Sarah Bolton
sarah.bolton@afit.edu

Extended author information available on the last page of the article

estimated to be about 21 zettabytes and is doubling every two years [2]. If we assume the average personal computer (PC) has a hard drive of one terabyte, 21 zettabytes is equivalent to 21 billion PCs, essentially three PCs for every person in the world. While a lot of this data is personal data, a large portion of it is considered publicly available information (PAI) and can be utilized by any internet user or organization.

This increase in available data has resulted in the study of identifying trends (i.e., data analytics), becoming more and more prevalent in multiple facets of society to include commerce and government. Researchers and major corporations have considered multiple ways to best utilize this massive resource, aptly referred to as 'Big Data.' Some areas that have shown promise include Internet of Things (IoT) analysis [3–5], traffic modeling [6], flight and maritime movement [7–11], image recognition [12], search engines [12] and natural language processing [12].

The increased focus on PAI and data analytics is recognized by military defense strategists who are responsible for making sound defense decisions. By incorporating PAI with the plethora of sensor data at their disposal, such as data from intelligence, surveillance, and reconnaissance platforms, it is possible to improve the predictive power of those resources. The need for data analytics is apparent in the United States Air Force and Space Force where multi-domain operations are integral to their defense strategies. In fact, the FY22 Posture Statement calls out Command and Control's need for the translation and sharing of data to provide 'real-time dissemination of actionable information' to provide 'joint warfighting across all domains at a pace faster than our competitors' [13]. Without recent advances in technology, artificial intelligence, and machine learning, this goal would be virtually impossible. Fortunately, new techniques can be used to filter the noise in big data much faster than human speed to quickly make inferences that are important to military decision makers.

To aid military leaders with analyzing the immense data at their disposal, we seek to improve military operations by providing enhanced capabilities for a major user of big data: intelligence analysts. One focus area important to intelligence analysts is pattern-of-life (POL) modeling. Some researchers seek to improve POL modeling via machine learning [14–18]. Recent research interests suggest analyzing ground-based and onboard aircraft sensors with deep learning to predict aircraft characteristics.

One stream of research for POL modeling is focused on exploiting automatic dependent surveillance-broadcast (ADS-B) data to make predictions about aircraft [6, 8, 11, 19]. Aircraft within certain airspace are required to broadcast ADS-B Out via an onboard transponder. The benefit of using ADS-B data for classification problems is that it is publicly available and aircraft flying in the USA and Europe are required to broadcast it in most classes of airspace [20, 21]. ADS-B data is collected from various sites worldwide where hobbyists and researchers maintain a receiver to collect it. ADS-B collectors submit their data to centralized repositories, such as the ADS-B Exchange [22], that aggregate the data for public use. In these repositories, both statistical and kinematic information about the broadcasting aircraft is available.

1.1 Problem description/objective

Pattern-of-life (POL) modeling is a research area with many techniques and best practices [14–18]. Military and defense personnel have an interest in POL modeling that includes more than modeling human day-to-day activities. For example, unattributed data from aircraft sensors, such as those collected from Air Traffic Control's (ATC) primary radar, can allow inferences to be made about the transmitting aircraft with some analysis. ATC's primary radar collects kinematic information such as location and airspeed, but is unable to obtain an aircraft's identification without the aircraft providing it via its transponder. With this basic kinematic aircraft data, models can predict information such as aircraft model or engine type without it being directly stated in the original dataset. The benefit of ADS-B data is that these features are present in the dataset and can be used as truth data to build models for datasets that do not have the truth data.

Since this type of processing can be resource intensive, it can be difficult or, in some cases, impossible to train a deep learning model when dealing with limited computing resources. The amount of computing resources required to train a model is heavily influenced by the size of the training data. For this reason, it is important to understand how to best utilize the available resources by minimizing the data used to train the model. There are two ways to minimize the data: limit the number of features or reduce the number of training samples. In this research, using aircraft kinematic data, we examine the impact of varying these factors when predicting engine type. Since reducing the training data will inevitably reduce the accuracy of the resulting model, for the purpose of this paper, we define an acceptable model as one that predicts within 10% of the previous baseline research results of 89.2% accuracy [23]. Therefore, models that can achieve at least a 79.2% accuracy will be considered 'acceptable.'

1.2 Research contribution

The research contribution of this paper can be summed up in the following points:

- Since there is no definitive guideline for minimum dataset size for deep learning classification problems, this research aims to determine a baseline for aircraft prediction models.
- This paper determines the baseline features to identify an aircraft with kinematic data: Speed, barometric pressure, and vertical speed
- This paper analyzes and reiterates the importance of selecting appropriate features. The 'noise' feature within this dataset severely limited the classification power of the network.

1.3 Organization

The rest of this paper is organized as follows: An exhaustive literature review and background information on ADS-B is provided in section two. In the third section, the methodology and process used to develop and evaluate each model is discussed. The fourth section presents the results. The conclusion is provided in the fifth section.

2 Background and literature review

This section describes automatic dependent surveillance-broadcast (ADS-B) data; previous attempts to classify engine type from it; relevant deep learning techniques; and the recommendations for dataset size when building a neural network model. Table 1 outlines the papers discussed in this section.

2.1 Automatic dependent surveillance-broadcast (ADS-B)

Automatic dependent surveillance-broadcast (ADS-B) is an aircraft sensor used throughout many regions of the world. Several researchers have utilized ADS-B data to identify flight patterns [8, 11, 24, 25], improve aircraft operations [26, 27], increase ADS-B transmission security [28, 29] and classify targets [7]. The transmissions are openly broadcast via an ADS-B Out transponder at 1090 MHz in many countries worldwide and at both 1090 MHz and 978 MHz within the USA. ADS-B data is received and collected via various methods to include other aircraft, ground stations, and satellites. Figure 1 depicts an ADS-B communications network.

Ground stations consist of sites where antennas collect ADS-B transmissions from passing aircraft. The ground stations are typically managed by commercial or government organizations, but hobbyists also collect the broadcasts. In the case of hobbyists, the aircraft transmissions are collected on a nearby device and transferred to one or multiple organizations that host the data for public use (e.g., the ADS-B Exchange [22]).

The ADS-B Exchange and similar services save the incoming data to their servers in two- to five-second intervals as a JavaScript Object Notation (JSON) file. Per the USA's DO-260B standard and Europe's ED-102A standard, each broadcast contains up to 70 features about the transmitting aircraft including the International Civil Aviation Organization (ICAO) identifier, altitude, airspeed, vertical speed, directional heading, position time, latitude, longitude, and ground status [22, 30].

Table 2 provides details on the countries that have an ADS-B mandate and the date when the mandate began or is projected to start. The chart shows that most countries, including the USA, Australia, the European Union Aviation Safety

Table 1 Comparative study table

| Paper | ADS-B | Prediction & classification | Improve aircraft operations | Transmission security | Dataset size | Collection & processing |
|-------------------------------|-------|-----------------------------|-----------------------------|-----------------------|--------------|-------------------------|
| Ginoulhac et al. (2019) [7] | ✓ | ✓ | | | | |
| Qian et al. (2019) [8] | ✓ | ✓ | | | | |
| Kumar et al. (2021) [11] | ✓ | ✓ | | | | |
| Basrawi (2021) [19] | ✓ | ✓ | | | | |
| Basrawi et al. (2021) [23] | ✓ | ✓ | | | | |
| Sun et al. (2016) [24] | ✓ | ✓ | | | | |
| Sun et al. (2017) [25] | ✓ | ✓ | | | | |
| Ruseno et al. (2022) [26] | ✓ | | ✓ | | | |
| Filippone et al. (2021) [27] | ✓ | | ✓ | | | |
| Hasin et al. (2021) [28] | ✓ | | | ✓ | | |
| Pearce et al. (2021) [29] | ✓ | | | ✓ | | |
| Sun (2021) [30] | ✓ | | | | | ✓ |
| Karim et al. (2017) [31] | | ✓ | | | | |
| Karim et al. (2019) [32] | | ✓ | | | | |
| Goodfellow et al. (2016) [33] | | ✓ | | | | |
| Hu et al. (2018) [34] | | ✓ | | | | |
| Alwosheel et al. (2018) [35] | | | | | ✓ | |
| Cho et al. (2015) [36] | | | | | ✓ | |
| Jain et al. (1982) [37] | | | | | ✓ | |
| Baum et al. (1988) [38] | | | | | ✓ | |
| Haykin (2009) [39] | | | | | ✓ | |

Agency (EASA), and many East Asia and Pacific island countries, began their mandate on or before 2020 which forces aircraft that fly within those regions to install the appropriate ADS-B Out equipment.

Within the USA, the mandate is not required for low flying aircraft and aircraft operating out of rural airports. Table 3 explains the US requirements in more detail. Within the USA, ADS-B Out is required for aircraft flying above 10,000 feet MSL

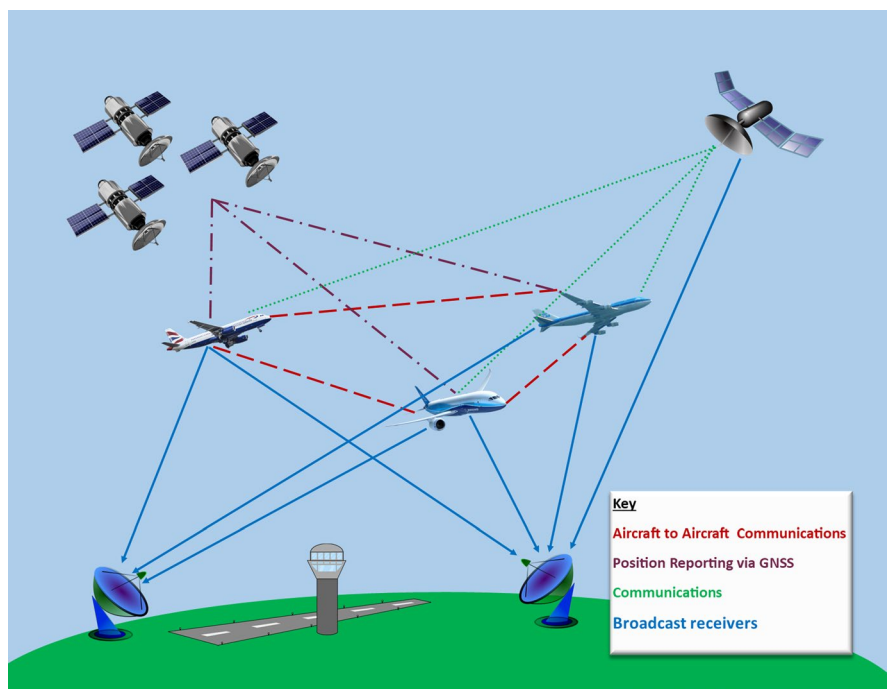


Fig. 1 ADS-B high-level operational concept graphic

and aircraft operating in locations near airports classified by Class B or Class C airspace.

2.2 Multivariate long short-term memory–fully convolutional neural networks

Artificial neural networks (ANNs) have been shown to be successful at classifying a variety of datatypes. Many ANN variations have been developed over the years to classify the infinite number of datatypes researchers encounter. Long short-term memory (LSTM) networks were developed to classify time series data and convolutional neural networks were developed to classify images. Combining these two networks, the multivariate long short-term memory–fully convolutional network (MLSTM–FCN) is shown to improve upon previous methods to classify time series data [31]. The algorithm, developed by Karim et al., combined fully convolutional networks (FCN), LSTM networks, and squeeze-and-excite blocks. FCNs were utilized to allow for the CNN benefit of class action maps without the requirement of extensive hyperparameter preprocessing that is normally required by a CNN. LSTMs were selected to detect the importance of sequences of observations in time series data. Squeeze-and-excite blocks were added to the algorithm to improve the classification power of multivariate datasets by ensuring feature maps have a similar impact to subsequent layers. In their study, Karim et al. tested four variations of the algorithm against 35 different datasets to include voice, human signal monitoring,

Table 2 ADS-B mandate timelines [40]**8. 1 Global ADS-B Out mandate summary**

| Mode S Transponders with Extended Squitter capability in accordance with RTCA DO-260 or DO-260B: | |
|--|------------------------------|
| Australia mandate for retrofit aircraft | 12-Dec-2013 |
| Singapore mandate for all aircraft | 12-Dec-2013 |
| Indonesia mandate for all aircraft | 01-Jan-2018 |
| Hong Kong mandate for all HKG-registered aircraft | |
| - Flying PBN routes L642 or M771 between FL290 and FL410 | 31-Dec-2013 |
| - Flying within HKG FIR between FL290 and FL410 | 31-Dec-2014 |
| Australia mandate for SA Aware GNSS (new a/c) | 08-Dec-2016 |
| Mexico - SBAS GPS required as per FAA guidelines | January 1, 2020 (Pending) |
| Colombia - SA-ON acceptable | January 1, 2020 (Pending) |
| U.S. FAA mandate for 100% equipage | 01-Jan-2020 |
| - Class A, Class B and Class C airspace | |
| - 1090 ES (Extended Squitter) for FL180 and above | |
| - 1090 ES or UAT (Universal Access Transceiver) below FL180 | |
| - Must be compliant with TSO C166b (Mode S) or C154a (UAT) | |
| EASA mandate for forward-fit and retrofit aircraft | 07-June-2020 |
| China mandate for all aircraft | 31-Dec-2022 |

and other time series data. Their algorithms outperformed the current state-of-the-art algorithm in 27 out of the 35 datasets [31]. Additionally, a variation of the algorithm, the multivariate attention long short-term memory–fully convolutional network (MALSTM–FCN), predicted aircraft engine type using ADS-B data [19]. This research will be further discussed in the next section.

While LSTMs and FCNs are well established, the use of squeeze-and-excite blocks is a much newer technique. To understand how they improve the MLSTM–FCN, it's important to understand how they work and what they provide. Squeeze-and-excite blocks were introduced by Hu et al. in 2018 to improve the representational capacity of a neural network [34]. They are used with convolutional networks to help model the dependencies between the channels. As the name implies, it consists of a squeeze mechanism followed by an excitation mechanism. In the squeeze step, the classifier uses global average pooling to aggregate the spatial information of each channel. Using the example of an image, where an image has dimensions of $H \times W \times C$ for height, width, and (normally 3) channels, the squeeze step would pass the image through a global average pooling operator where it would become a shape of $(1 \times 1 \times C)$. Then, the excite step uses a gating mechanism to capture channel-wise dependencies [32]. The excitation step uses a multi-layer neural network with one hidden layer. The input and output layer are the same shape $(1 \times 1 \times C)$, but the hidden layer reduces the space by a reduction factor, r , making the number of neurons in the hidden layer C/r . Karim et al. and Hu et al. use a reduction factor of 16 [32, 34]. This $(1 \times 1 \times C)$ value is multiplied element-wise with the original $(H \times W \times C)$ input. The graphical representation of the squeeze-and-excite block developed by Hu et al. can be seen in Fig. 2.

Using the squeeze-and-excite block in conjunction with an LSTM and FCN, the MLSTM–FCN was developed to create a model for time series classification. Figure 3 shows the architecture Karim et al. designed for their research. This algorithm is used to develop all of the models in this research.

2.3 Dual-stage deep engine classifier

Multiple researchers have focused on using the ADS-B dataset to predict engine type with ADS-B kinematic data [8, 19]. One of the limitations with analysis on this dataset is that while the models have very few problems determining if the aircraft is a jet, they tend to have trouble predicting the difference between turboprop and piston engines. The reason for this is twofold. First, the dataset is heavily imbalanced toward jet engines which causes the data to have fewer samples of turboprop and piston engines from which to learn. Second, the kinematic differences between piston and turboprop engines are minimal.

As a method to remedy this problem, Basrawi et al. developed a Dual-Stage Deep Engine Classifier (DSDEC) [19]. Using the MALSTM–FCN algorithm as a basis for the model, the DSDEC algorithm employs a unique 2-stage approach. When using the model to make predictions, the first stage predicts if the aircraft is a jet or not a jet. Then, the ‘not jet’ predictions are fed to the second stage. The second stage predicts if the aircraft has a piston or turboprop engine. The results from both the first and second stage are combined to provide an engine prediction for each observation. Figure 4 portrays the architecture that is used.

Basrawi et al. are the first to classify aircraft engine type using ADS-B data with a deep learning model. Using the DSDEC method, researchers were able to identify jets with 98.4% accuracy, turboprop aircraft with 79.2% accuracy, and piston engine aircraft with 89.9% accuracy. However, the DSDEC method would still confuse turboprop aircraft as piston engine aircraft 17.9% of the time [19]. Table 4 shows the results achieved by Basrawi et al. compared to a support vector machine (SVM) and random forest (RF) as baseline from previous research[8].

During the experiments, static time steps were used and each time step was separated by two seconds. 300 time steps would be equivalent to 600 seconds or 10 minutes of flight time. Similarly, 100 time steps would be 200 seconds or 3 minutes and 20 seconds of flight time. While Basrawi et al. indicated that more research would be needed to determine the influence of the time step size, their results point to the possibility that longer flight observations improve the model’s predictive power. In fact, while the DSDEC algorithm was used against 21 different models to determine which hyperparameters led to the highest accuracy, none of the 100 time step models performed as well as the 300 time step examples.

2.4 Size of dataset

Determining the exact dataset size needed to build an artificial neural network (ANN) classifier is an open research problem that may never have a complete

solution due to the infinite number of ANN combinations. However, experts suggest several guidelines to ensure enough data is available to train a model with sufficient accuracy value [35–39]. Guidelines can be broken into three important characteristics of the dataset: the number of prediction classes, the number of features, and the number of weights in the network. The following guidelines sum up the results from past research endeavors:

1. **Prediction Classes:** A sample size should have 50–1000 times the number of observations as prediction classes [36]. For this paper, there are three prediction classes: Jet, turboprop, and piston.
2. **Features:** There should be 10–100 times the number observations as features [37].
3. **Network Weights:** The sample size should be equivalent to 10 times the number of weights in the network [38, 39].
 - (a) Another paper decided on a stricter limit stating that there should be 50 times the number of observations as network weights [35].

2.5 Summary

The information provided in this section shows that ADS-B sensor data is a useful resource when trying to understand POL modeling with aircraft. Previous research has used this data to predict aircraft characteristics [7, 8, 19]. When predicting engine type, researchers found that the MLSTM–FCN was able to train a model that achieved an overall accuracy of 89.2% [19]. This paper uses the information gleaned

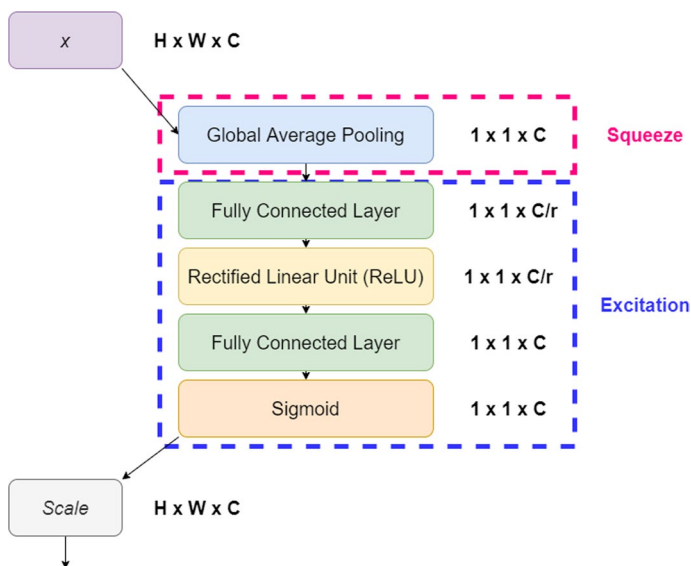


Fig. 2 The schema of the squeeze-and-excite model

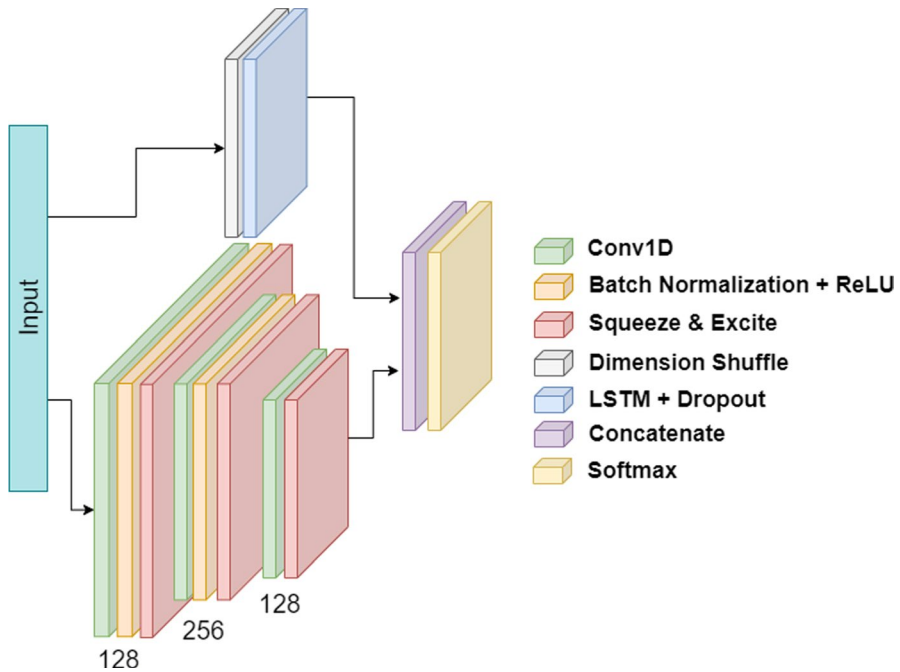


Fig. 3 MLSTM-FCN architecture [32]

Table 3 US ADS-B Airspace Requirements [20]

| Airspace | Altitude |
|-------------|--|
| Class A | All |
| Class B | Generally, from surface to 10,000 feet mean sea level (MSL) including the airspace from portions of Class Bravo that extend beyond the Mode C Veil up to 10,000 feet MSL (e.g. LAX, LAS, PHX) |
| Class C | Generally, from surface up to 4,000 feet MSL including the airspace above the horizontal boundary up to 10,000 feet MSL |
| Class E | At and above 10,000 feet MSL over the 48 states and DC, excluding airspace at and below 2,500 feet AGL Over the Gulf of Mexico at and above 3,000 feet MSL within 12 nautical miles of the coastline of the United States |
| Mode C Veil | Airspace within a 30 NM radius of any airport listed in Appendix D, Section 1 of Part 91 (e.g. SEA, CLE, PHX) from the surface up to 10,000 feet MSL |

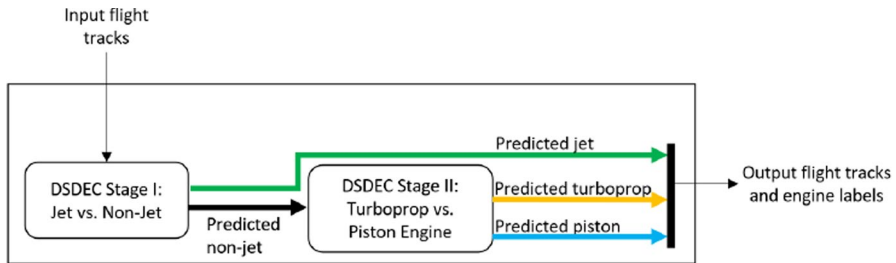


Fig. 4 DSDEC architecture [23]

Table 4 Results from Basrawi et al. [19]

| Classifier | Overall accuracy | Jet Accuracy | Turboprop accuracy | Piston engine accuracy |
|------------------------|------------------|--------------|--------------------|------------------------|
| DSDEC (300 time steps) | 89.2% | 98.4% | 79.2% | 89.9% |
| SVM (300 time steps) | 77.4% | 85.5% | 72.2% | 74.6% |
| SVM (100 time steps) | 68.6% | 81.1% | 56.6% | 68.1% |
| RF (300 time steps) | 83.4% | 94.1% | 70.5% | 85.8% |
| RF (100 time steps) | 76.7% | 90.0% | 61.5% | 78.6% |

Bold italic values indicate the best performing examples

from research done with the MLSTM-FCN [32] and DSDEC [19] to learn how to minimize the input data size while maintaining a similar accuracy and loss.

3 Methodology

This section outlines a method to efficiently classify engine type from ADS-B data. It improves on Basrawi et al [19] by reducing the model complexity and decreasing the requisite ADS-B dataset size to classify engine. Basrawi et al. used two feature sets, a limited and a full feature set, which had 6 and 9 features, respectively. Additionally, they used 9 days worth of ADS-B data to train a dual-stage classification algorithm. This method reduces the model complexity to a single stage and only requires a 24-hour sample size and three ADS-B features to achieve similar classification accuracy results. It uses data from 1 December 2020 to train the model and 16 November 2020 to test the model.

The rest of this section describes two experiments:

- **Experiment 1:** The first experiment creates 24 models with two different learning rates, three feature sets, and four data amounts. The models train over a period of 200 epochs and are evaluated on the testing data. The goal of the

first experiment is to determine the effect of varying the number of training data observations. Effects to accuracy, loss, precision, and recall are recorded.

- **Experiment 2:** In the second experiment we develop models using all possible feature combinations (255). The goal of the second experiment is to determine which subset of features has the highest overall accuracy when predicting engine type. The models train for 50 epochs, which is where the first experiment shows training and validation data accuracy diverge.

The details of each experiment can be seen in Table 5.

3.1 Assumptions and limitations

The dataset used in this research is smaller than the one used by Basrawi et al. [19]. In this research, the training data from 1 December 2020 has 4,110 tracks/tensors, and the evaluation data from 16 November 2020 has 2,487 tracks/tensors. In the experiment completed by Basrawi et al., the training data ranged from 1 to 8 December 2020 consisting of 7,749 tracks/tensors. The evaluation data was also from 16 November 2020, but consisted of 4,158 tracks. This amounts to approximately half of the tracks for training and evaluation in this experiment. It is assumed that the effects of reducing the data size would be proportional regardless if one week or one day was used for training.

Although researchers have shown that the integrity of ADS-B sensor data is vulnerable to a variety of cyber attacks [41, 42], the dataset is assumed accurate for the purposes of this paper. While vulnerable, cyber attacks against ADS-B are not a common occurrence. Air traffic control-related attacks occur only a few times each year [43]. We consider this to be an acceptable assumption since the number of occurrences of message injects and other ADS-B attacks is low in comparison to the millions of flights that occur each year.

3.2 Process

The process for converting raw ADS-B data into a model that can predict engine type can be broken down into five steps that will be explained in the subsequent paragraphs Fig. 5.

3.2.1 Data preparation

The ADS-B data acquired by Basrawi et al. [19] contains aircraft observations from thousands of locations each day from November to December 2020. During the preparation phase, the data is converted from JavaScript Object Notation (JSON) files to Python Data Analysis Library (Pandas) data frames, sorted by unique International Civil Aviation Organization (ICAO) and time, and then saved as Comma-Separated Values (CSV) files. Then, the CSV files are reloaded as a Pandas data frames with invalid and irrelevant data (e.g., helicopter and glider data points)

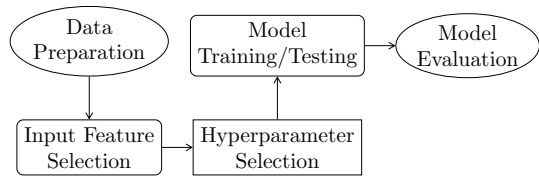
Table 5 Experiment Details

| Exp # | Goals | # of Models | Parameters |
|-------|---|-------------|--|
| 1 | Determine the effect of varying the amount of training data samples | 24 | Learning rate: 0.01 & 0.001 Drop out: 0.5 Time steps: 300 Feature set: limited, medium & full Optimizer: Adam Data size: Full, 1/2, 1/4, & 1/8 Epochs: 200 |
| 2 | Determine the effect of each subset of features | 255 | Learning rate: 0.001 Drop out: 0.5 Time steps: 300 Feature set: all possible subsets Optimizer: Adam Data size: Full Epochs: 50 |

removed. Aircraft without at least 300 transmissions (10 minutes) are removed and any transmissions after 300 time steps are discarded to keep the data as similar as possible. For all aircraft, only the first 10 minutes of the takeoff segment of flight are used. The remaining data is balanced by engine type using an undersampling technique [44, 45] to reduce the impact of the heavily imbalanced dataset. The undersampling technique is selected over an oversampling or a hybrid method since the dataset is already very large and diverse. Thousands of ‘Jet’ observations remain in the dataset even with undersampling. Since they are randomly removed, overfitting is avoided. Then, the engine type feature is converted from a number to a one hot encoding of Boolean values with ‘jet,’ ‘turboprop’ and ‘piston’ as possible options. The data preparation step reduces 44 GB of raw data to 270 MB of processed data.

The processed data is formed into (n, t, v) tensors [46], where n is the number of aircraft tracks, t is the number of time steps for each track, and v is the number of features describing each track. In the previous version of this experiment t and v were varied across multiple trials to determine the best combination. It was determined that reducing the t variable lower than 300, lowers the accuracy. Therefore, $t = 300$ is maintained across all iterations of this experiment. n and v are altered by changing the amount of training data and changing the number of features, respectively.

Experts agree on best practices to follow when training classification models. The first recommendation is that the training dataset sample size be at least 50-1000 times the number of prediction classes [36]. Since there are only three prediction classes and over a million observations, this goal is met. The second suggestion is that the sample size is 10-100 times larger the number of features [37]. The feature set with the most number of features is 12. Since $12 \times 100 = 1200$ and there are 1,233,000 samples, this suggestion is also met. The final guideline states that the number of observations should be 10 times the number of weights in the model.

Fig. 5 Data Processing Steps

Since the model has 280,000–288,000 weights (depending on the number of features for that experiment), the number of samples needed is near 3 million [38, 39]. With less than half the requisite samples, this guideline is not met. However, the study that this experiment was built on only used 2,324,700 samples and was able to achieve a nearly 90% accuracy [19]; it is assumed that not meeting the 10x weight requirement, but meeting all other guidelines, is sufficient for this experiment.

In the final part of data preparation, we create four training datasets of varying sizes. The first training dataset includes observations from the entire 24-hour period on 1 December 2020. This dataset is the largest dataset and is referred to as the full dataset throughout this paper. In the creation of the next three datasets, we randomly remove tensors from the full dataset to create a new dataset that is half, a quarter, and an eighth the size of the full dataset. Since it is done randomly, the datasets are not equal to 12, 6, and 3 continuous hours of observations. Instead, the observations are 300 time step tensors taken at various points during the 24-hour time period. Proportions between engine types are maintained per the undersampling technique. The full dataset results in 4,110 tensors (1,233,000 samples), the half dataset has 2,055 tensors (616,500 samples), the quarter dataset has 1,026 tensors (307,800 samples), and the eighth dataset has 513 tensors (153,900 samples). Table 6.

3.2.2 Input feature selection

The raw ADS-B data collected during November and December 2020 contained a total of 57 features. Most of them were either not consistently transmitted by all aircraft or contained irrelevant or redundant information. There were also a few features that contained identifying information like ICAO, aircraft model, ID number, country of origin, inbound/outbound location, or other non-kinetic information that is not intended to be used for this research. This reduced the usable kinetic input features from 57 to 9. We also develop 3 other input features from these inputs to improve the location data. The definition of each feature is listed in Table 7.

The following features are selected for inclusion into the dataset due to their potential for predicting engine type.

1. Altitude and Ground Altitude—Since jet, turboprop, and piston engine aircraft tend to fly best at different altitudes, these features are important in distinguishing between them. [48]
2. Airspeed—Jets fly faster than piston or turboprop engines. Turboprop engines can reach greater speeds easier at higher altitudes than piston engines. [48]

3. Barometric Pressure—This feature is another way to measuring altitude. For every thousand feet of elevation, the pressure drops by 1 inHg. [49]
4. Vertical Speed—The importance of this feature is similar in nature to air/ground speed.
5. Time—This feature would allow the comparison of different chronological points of the flight
6. Track—With location and speed, track can be used to learn specific aircraft patterns.
7. Lat/Long—Aircraft may exhibit different behaviors depending on the geography. For example during the first ten minutes, an aircraft would takeoff differently from a mountainous region than an open field.
8. Location (X,Y,Z)—Lat/Long are normalized to better represent a 3-dimensional space. This feature was not in the raw data, but was instead generated to help better represent the data. The formulas used to normalize the lat/long are as follows:

$$(a) \quad x = \cos(\text{lat}) * \cos(\text{lon})$$

$$(b) \quad y = \cos(\text{lat}) * \sin(\text{lon})$$

$$(c) \quad z = \sin(\text{lat})$$

Feature combinations for the first experiment are tested based on their size and their possible effects on classification. The feature combinations experiment one uses are outlined in Table 8.

For the second experiment, all possible combinations of the full feature set are used to determine the best subset of features. However, latitude and longitude are not used for this experiment. They are replaced with the normalized location coordinates.

3.2.3 Hyperparameter selection

Since hyperparameters were already tested by Basrawi et al. to determine the best combination, this study uses those hyperparameters for the model creation with the exception of learning rate. Basrawi et al. found that the Adam optimizer and learning rates of 0.01 and 0.001 performed the best, while dropout rates were mostly inconsequential. Based on their findings, we train the model with a dropout rate of 0.5 and learning rates of 0.01 and 0.001 with the Adam optimizer.

Table 6 Dataset details from 1 december 2020

| Size | Samples | Tensors |
|---------|----------|---------|
| Full | 1,233,00 | 4,110 |
| Half | 616,500 | 2,055 |
| Quarter | 307,800 | 1,026 |
| Eighth | 153,900 | 513 |

Table 7 ADS-B Data Features/Fields [22, 47]

| Feature | Field | Data type | Description |
|---------------------|---------|-----------------|--|
| Altitude | Alt | integer | The altitude in feet at standard pressure |
| Ground Altitude | Galt | integer | The altitude adjusted for local air pressure |
| Airspeed | Spd | knots (float) | The ground speed in knots |
| Barometric Pressure | InHg | float | The air pressure in inches of mercury that was used to calculate the AMSL altitude from the standard pressure altitude |
| Vertical Speed | Vsi | integer | Vertical speed in feet per minute |
| Time | PosTime | epoch (ms) | The time that the position was last reported by the aircraft. |
| Track | Trak | degrees (float) | Aircraft track angle across the ground clockwise from 0 north. |
| Latitude | Lat | float | The aircraft's latitude over the ground |
| Longitude | Long | float | The aircraft's longitude over the ground |
| Location | X,Y,Z | float | Cartesian coordinates of the aircraft. This data feature is derived from the lat/long coordinates and it is not originally part of the ADS-B broadcast |

3.2.4 LSTM training, testing, and evaluation

Similar to Basrawi et al. [19], this study uses the algorithm developed by Karim et al. [31]. The two major differences are that this research omits the attention mechanism and does not use the two-phase approach suggested by Basrawi et al. [19]. In the first experiment, 24 separate models are created out of the data from 1 December 2020 to encompass the variations in learning rate (0.01 and 0.001), the feature set (limited, medium, and full), and the data amount used (24 hours, half of the data, a quarter of the data and an eighth of the data). Each model is trained for 200 epochs which in most cases is more than sufficient.

In the second experiment, 255 models are created using a 0.001 learning rate and trained on the full size dataset. The features vary between each model to evaluate all possible combinations of features. Each model is trained for 50 epochs which is the point when training and validation data accuracy deviate.

Accuracy, precision, loss, and recall are saved during the creation of the models for both experiments to show the models' histories during each epoch. Those metrics are also saved for the final models' evaluation. A k -fold method, where $k = 10$, tests each model with the data from 16 November 2020. Each model is compared to show how the change in both the input feature size and the amount of data points affected the aforementioned metrics.

4 Results and discussion

4.1 Experiment 1–24 models: data size comparison

Table 9 outlines the performance across all phases. Figures 6 and 7 represent the data from Table 9. The line colors represent the input feature size and the color's shade represents the learning rate. The 0.001 learning rate is darker shade and the 0.01 learning rate is the lighter shade. Blue represents the smallest input feature size, green the medium input feature size, and orange the full input feature set. As can be seen, the limited feature set has the best accuracy and loss with the medium and full feature set having very irregular performance. While not shown, recall and precision follow similar trends.

Model 1 performs the best. It has an 89.4% overall accuracy which is on par with the 89.2% accuracy achieved by Basrawi et al. [23], but with half the sample size. The confusion matrix for model 1 can be seen in Figure 8. The results for individual engine type are also similar to those collected by Basrawi et al. For comparison, they found that jet engines were predicted correctly 98.4% of the time, turboprop 79.2%, and piston 89.9% [23]. In this study, jets are accurately predicted 97.2% of the time, turboprops are 79.1% and pistons are 92.0%.

The results show that the limited feature dataset outperforms the medium and full feature sets. The larger feature sets never converge. This phenomenon is due to noise and overfitting to the training data. Experiment two is able to further analyze the lack of convergence. However, using the three feature sets created for this experiment, it is shown that altitude, airspeed, vertical speed, and location provide sufficient information about aircraft to differentiate engine type in most cases. The additional features do not provide any information that help the model learn more about aircraft engines. Instead, it makes the data more convoluted. This can be shown with the training history as seen in Figures 9, 10, and 11. These figures represent the training accuracy after each epoch where the feature set is varied between the figures, but learning rate and dataset size remain constant. The three images are from the models that had a 0.001 learning rate and used the full dataset (i.e., models 1, 3, and 5). While the training accuracy continues to increase, the validation accuracy does not. This indicates that the model is no longer improving and is instead overfitting to the training data. While only a subset of the models are shown in this manner, other models follow a similar pattern with the limited feature set performing the best in all cases.

The results from the limited feature set show that more data improves the classification power of the model. However, the reduction in accuracy is

Table 8 Experiment Trial Setup

| Limited feature set | Medium feature set | Full feature set |
|--|--|--|
| Altitude Airspeed Vertical speed Location (X,Y,Z) | Altitude Airspeed Vertical speed Location (X,Y,Z) Barometric Pressure Time Track | Altitude Airspeed Vertical speed Location (X,Y,Z) Barometric Pressure Time Track Ground Altitude Lat/Long |

gradual until the one-eighth size dataset. Based on our previous definition of an ‘acceptable’ model, the 1/4 size dataset meets the minimum accuracy with 83.9%. For this reason, if computational power was limited, someone could drop the number of inputs to 1/4 of the full dataset and still be guaranteed to have a reasonable prediction accuracy rate.

4.2 Experiment 2–255 models: feature set comparison

The goal for experiment two is to find the subset of features that best predict engine type from the full size dataset. All possible feature combinations from 8 available features result in the creation of 255 models. The top 20 results from the 255 models are presented in Table 10. Airspeed, pressure, and vertical airspeed create the best-performing dataset. This result is unexpected since it does not include altitude. Jets fly at a different altitude than piston or turboprop engines. For that reason, it would seem like altitude would be an important feature. A possible reason for this might be that the observations only include the first 10 minutes after takeoff. Jet engines would not have the time to reach cruising altitude until the end of the 10 minute time frame.

Additionally, the results from experiment show that a combination of all of the features except for ‘PosTime’ provide the third best feature indicator for accuracy. However, not only is ‘PosTime’ not included in the 3rd best combination, but it is also interesting to note that all of the worst-performing models contain ‘PosTime.’ Table 11 shows the 10 worst-performing models. It is not until the 43rd worst performer (the model with just the ‘Trak’ feature with a 48.9% accuracy rate) that the ‘PosTime’ feature is not present.

5 Conclusion

The goal of this research is to determine the effect of minimizing the size of the training data that was used to develop a MLSTM–FCN model. We do this with two separate experiments. In experiment one, we vary the number of training samples in the training dataset with three different feature sets. In experiment two, we vary the number of features present in the dataset. Those models are tested to determine how each change affects the accuracy and loss of the model against a separate test set. We deem a model to be ‘acceptable’ if the accuracy was within 10% of results from previous research [19].

There are a few main takeaways from this experiment. The first is that the quality of a dataset is more important than the quantity of data when building a neural network. Having redundant features or features that do not add pertinent information to the model can cause the model to perform poorly. This fact is convenient when it comes to minimizing a dataset, but can be frustrating when trying to determine

Table 9 Performance comparisons

| Num | Learn rate | Feature set | Data amt | Mean accuracy | Mean loss |
|-----|------------|-------------|----------|-------------------------|-------------------------|
| 1 | 0.001 | Limited | Full | 0.894 (+/-0.018) | 0.413 (+/-0.069) |
| 2 | 0.01 | Limited | Full | 0.890 (+/-0.019) | 0.346 (+/-0.083) |
| 3 | 0.001 | Medium | Full | 0.513 (+/-0.031) | 1.321 (+/-0.092) |
| 4 | 0.01 | Medium | Full | 0.337 (+/-0.019) | 3.604 (+/-0.117) |
| 5 | 0.001 | Full | Full | 0.759 (+/-0.038) | 0.675 (+/-0.100) |
| 6 | 0.01 | Full | Full | 0.356 (+/-0.018) | 2.960 (+/-0.210) |
| 7 | 0.001 | Limited | Half | 0.848 (+/-0.019) | 0.474 (+/-0.072) |
| 8 | 0.01 | Limited | Half | 0.865 (+/-0.023) | 0.367 (+/-0.051) |
| 9 | 0.001 | Medium | Half | 0.492 (+/-0.029) | 1.493 (+/-0.102) |
| 10 | 0.01 | Medium | Half | 0.490 (+/-0.031) | 1.622 (+/-0.102) |
| 11 | 0.001 | Full | Half | 0.740 (+/-0.025) | 0.595 (+/-0.053) |
| 12 | 0.01 | Full | Half | 0.354 (+/-0.023) | 3.071 (+/-0.149) |
| 13 | 0.001 | Limited | 1/4 | 0.839 (+/-0.016) | 0.473 (+/-0.098) |
| 14 | 0.01 | Limited | 1/4 | 0.839 (+/-0.014) | 0.478 (+/-0.054) |
| 15 | 0.001 | Medium | 1/4 | 0.700 (+/-0.026) | 0.789 (+/-0.078) |
| 16 | 0.01 | Medium | 1/4 | 0.391 (+/-0.040) | 2.211 (+/-0.152) |
| 17 | 0.001 | Full | 1/4 | 0.658 (+/-0.028) | 0.891 (+/-0.089) |
| 18 | 0.01 | Full | 1/4 | 0.505 (+/-0.040) | 1.645 (+/-0.121) |
| 19 | 0.001 | Limited | 1/8 | 0.758 (+/-0.019) | 0.897 (+/-0.124) |
| 20 | 0.01 | Limited | 1/8 | 0.688 (+/-0.037) | 1.525 (+/-0.224) |
| 21 | 0.001 | Medium | 1/8 | 0.678 (+/-0.028) | 1.008 (+/-0.088) |
| 22 | 0.01 | Medium | 1/8 | 0.607 (+/-0.026) | 1.397 (+/-0.175) |
| 23 | 0.001 | Full | 1/8 | 0.595 (+/-0.038) | 1.833 (+/-0.179) |
| 24 | 0.01 | Full | 1/8 | 0.484 (+/-0.021) | 2.120 (+/-0.116) |

Bold value indicate the best performing examples

which features are actually important. Too many irrelevant features will add noise into the model and produce results that are less than ideal. Based on this experiment, speed, pressure and vertical speed are some of the more important features needed to identify an aircraft's engine type.

This dataset is reduced to a quarter of its original size before the model starts exhibiting severe negative effects. During both experiments, the best model achieves an accuracy of 89.4%. Reducing the data by half only reduces the accuracy by 4.6% to 84.8% accuracy which has a rate of change of 9.2% (Δ accuracy/ Δ size). Reducing it to a quarter of its original size produces an 83.9% accuracy rate which is a 22.0% rate of change from the full size dataset. However, reducing the dataset to an eighth of its original size produces a 75.8% accuracy rate which is a 108.8% rate of change from the full size dataset. Since the raw data from 1 December 2020 was about 44 GB, this large file size could be reduced to 11 GB using the quarter size dataset and still effectively train a classification model. Most computers, laptops, or small handheld devices could perform data processing and train a model that was only 11 GB. This observation would work well for military operations in remote locations with low computational resources. If data and computation resources were

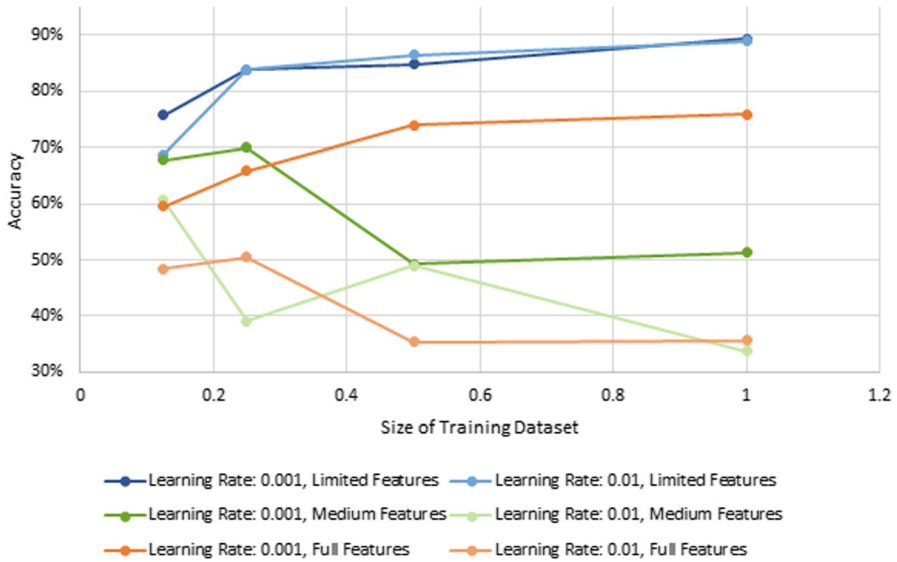


Fig. 6 Change in accuracy by change in sample size

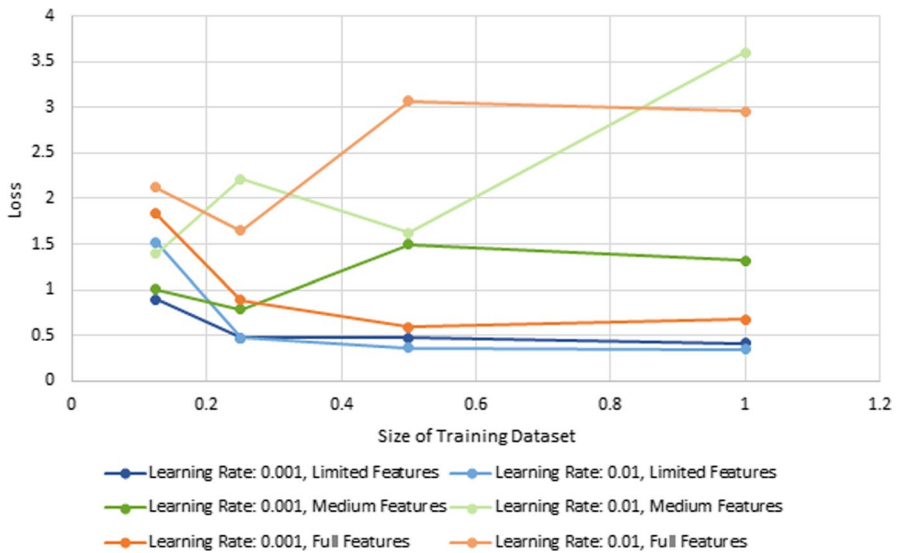


Fig. 7 Change in loss by change in sample size

not as plentiful, reducing the model to approximately 300,000 observations would create a model that could make observations with a small trade-off of less than 10% reduced accuracy.

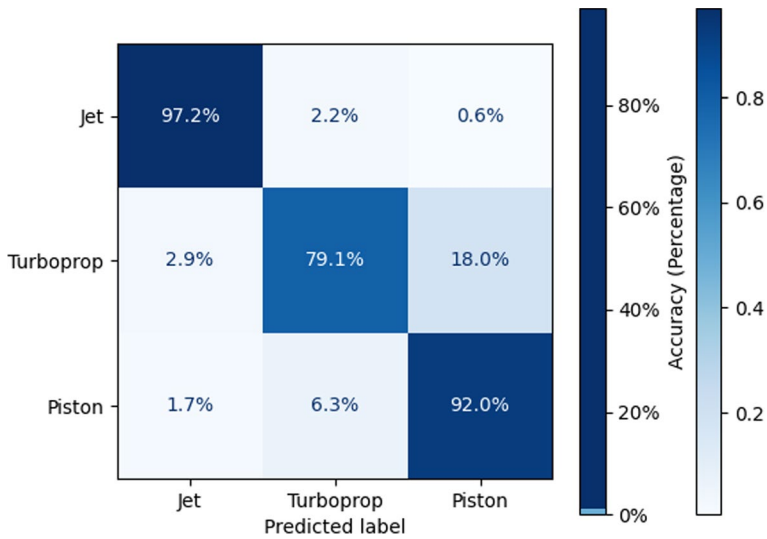


Fig. 8 Confusion Matrix for Model 1

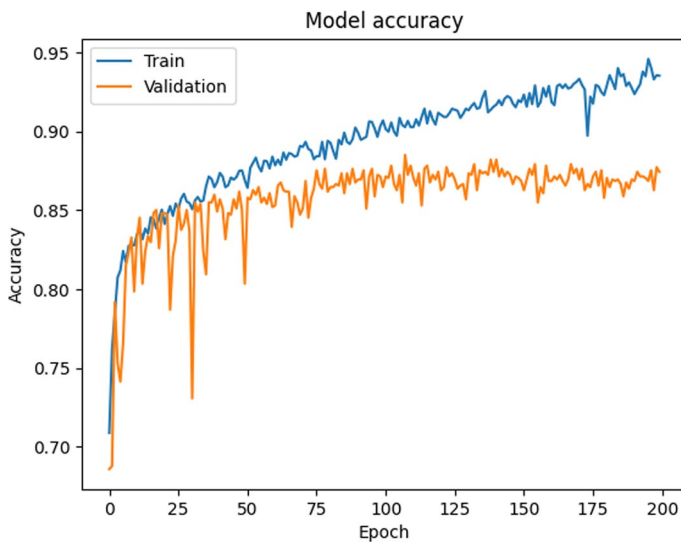


Fig. 9 Accuracy Training History on Limited Feature Set for 0.001 Learning Rate and Full Dataset

5.1 Future work

While the goal of this study is to determine how to minimize the input data to make best use of low computational resources, military operations tend to have access to a vast number of sensors. Combining other related sensors to the ADS-B data could improve results. Other potentially useful sensors include weather, radar, and image

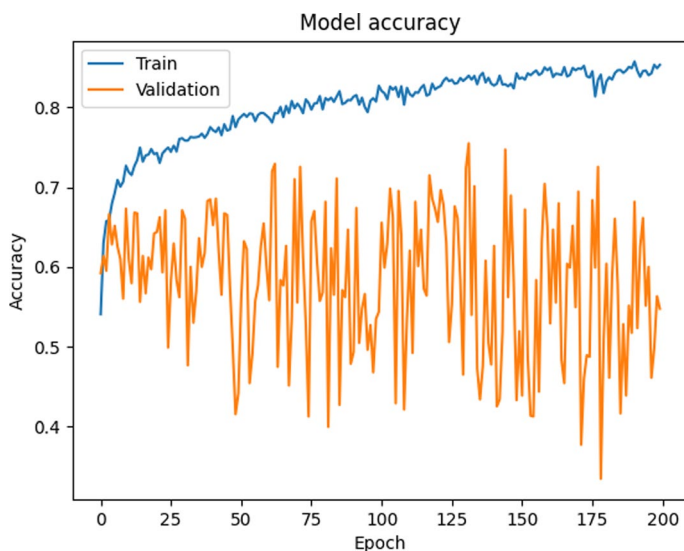


Fig. 10 Accuracy Training History on Medium Feature Set for 0.001 Learning Rate and Full Dataset

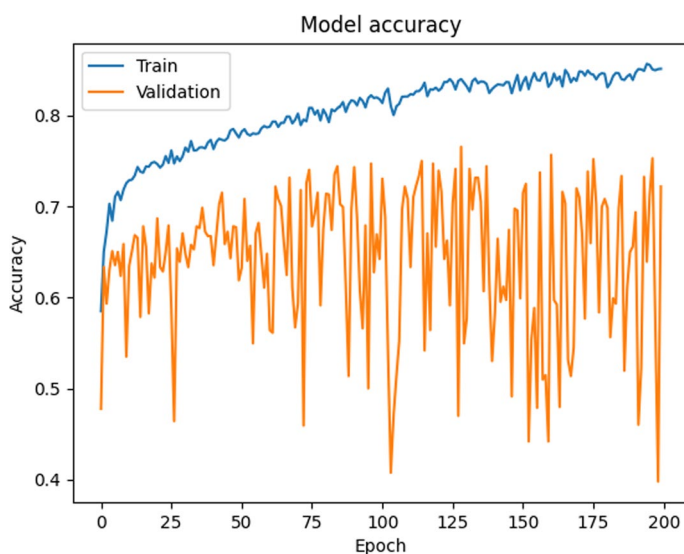


Fig. 11 Accuracy Training History on Full Feature Set for 0.001 Learning Rate and Full Dataset

data. Researchers should look at the effects of combining these sources to determine if they improve classification accuracy.

The raw ADS-B data represents the ‘PosTime’ feature as the number of milliseconds since Epoch (1970). During the processing of the data, ‘PosTime’ is modified to more closely compare to previous work with engine type prediction. Instead of milliseconds since Epoch, it represents the amount of milliseconds since takeoff.

Table 10 Top 20 Performance Comparisons

| Ex | Feature set | Mean accuracy | Mean loss |
|-----|---|---------------------|---------------------|
| 56 | Spd,InHg,Vsi | 0.894 (+/-0.013) | 0.304 (+/-0.038) |
| 40 | Spd,Vsi | 0.887 (+/-0.025) | 0.336 (+/-0.088) |
| 251 | Alt,Galt,Spd,InHg, Vsi,Trak,location | 0.886 (+/-0.023) | 0.319 (+/-0.056) |
| 59 | Spd,InHg,Vsi,Trak, location | 0.886 (+/-0.015) | 0.307 (+/-0.060) |
| 186 | Alt,Spd,InHg,Vsi,Trak | 0.884 (+/-0.017) | 0.330 (+/-0.059) |
| 171 | Alt,Spd,Vsi,Trak, location | 0.884 (+/-0.016) | 0.366 (+/-0.132) |
| 43 | Spd,Vsi,Trak,location | 0.884 (+/-0.015) | 0.326 (+/-0.066) |
| 42 | Spd,Vsi,Trak | 0.877 (+/-0.017) | 0.346 (+/-0.067) |
| 185 | Alt,Spd,InHg,Vsi, location | 0.876 (+/-0.026) | 0.318 (+/-0.056) |
| 105 | Galt,Spd,Vsi,location | 0.876 (+/-0.019) | 0.347 (+/-0.052) |
| 105 | Galt,Spd,Vsi,location | 0.876 (+/-0.019) | 0.347 (+/-0.052) |
| 184 | Alt,Spd,InHg,Vsi | 0.875 (+/-0.020) | 0.332 (+/-0.070) |
| 235 | Alt,Galt,Spd,Vsi,Trak, location | 0.875 (+/-0.018) | 0.362 (+/-0.060) |
| 58 | Spd,InHg,Vsi,Trak | 0.875 (+/-0.015) | 0.360 (+/-0.057) |
| 248 | Alt,Galt,Spd,InHg,Vsi | 0.874 (+/-0.017) | 0.342 (+/-0.056) |
| 123 | Galt,Spd,InHg,Vsi,Trak, location | 0.874 (+/-0.015) | 0.370 (+/-0.064) |
| 169 | Alt,Spd,Vsi,location | 0.872 (+/-0.020) | 0.364 (+/-0.077) |
| 41 | Spd,Vsi,location | 0.871 (+/-0.023) | 0.334 (+/-0.037) |
| 104 | Galt,Spd,Vsi | 0.871 (+/-0.020) | 0.369 (+/-0.060) |
| 114 | Galt,Spd,InHg,Trak | 0.871 (+/-0.017) | 0.368 (+/-0.062) |
| 106 | Galt,Spd,Vsi,Trak | 0.871 (+/-0.013) | 0.396 (+/-0.069) |

When creating a model with an LSTM, this is not the best use of the time feature. Keeping track of time of day, day of the week, or even just the date would allow the model to better learn aircraft schedules. Future work should look at modifying this feature into something that would improve the classifier.

Table 11 Bottom 10 performance comparisons

| Ex | Feature set | Mean accuracy | Mean loss |
|-----|---|------------------|--------------------|
| 7 | PosTime,Trak,location | 0.337 (+/-0.038) | 1.622 (+/-0.088) |
| 29 | InHg,Vsi,PosTime, location | 0.337 (+/-0.037) | 4.511 (+/-3.705) |
| 21 | InHg,PosTime,location | 0.337 (+/-0.028) | 38.038(+/-109.699) |
| 22 | InHg,PosTime,Trak | 0.336 (+/-0.024) | 1.615 (+/-0.117) |
| 52 | Spd,InHg,PosTime | 0.334 (+/-0.024) | 2.812 (+/-0.138) |
| 127 | Galt,Spd,InHg,Vsi, PosTime,Trak,location | 0.334 (+/-0.019) | 18.762 (+/-1.317) |
| 37 | Spd,PosTime,location | 0.332(+/-0.016) | 5.308(+/-0.238) |
| 55 | Spd,InHg,PosTime,Trak, location | 0.114 (+/-0.022) | 3.323 (+/-0.116) |
| 38 | Spd,PosTime,Trak | 0.110 (+/-0.018) | 2.742 (+/-0.072) |
| 54 | Spd,InHg,PosTime,Trak | 0.074 (+/-0.015) | 4.664 (+/-0.112) |

This research uses only the first 10 minutes of flight to train each model. The first 10 minutes of flight consists of the takeoff phase and sometimes part of the cruising phase. Incorporating other phases, such as cruising and landing, could help to further separate the differences between engine types. The biggest benefit of using other phases is that jets, turboprops, and pistons have different characteristics when they get to the cruising phase. Jets fly at much higher altitudes during the cruising phase than other engine types and turboprop engines can reach greater speeds easier at higher altitudes than piston engines [48].

Funding This study was funded by the Air Force Research Laboratory.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interests The authors have no relevant financial or non-financial interests to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bartley K (2021) Big data statistics: how much data is there in the world? . <https://riverty.io/blog/big-data-statistics-how-much-data-is-there-in-the-world/>
2. Live-counter.com: how big is the internet? <https://www.live-counter.com/how-big-is-the-internet/>
3. Saleem TJ, Chishti MA (2019) Deep learning for internet of things data analytics. *Proc Comput Sci* 163:381–390. <https://doi.org/10.1016/j.procs.2019.12.120>
4. Dholey MK, Sinha D, Mukherjee S, Das AK, Sahana SK (2020) A novel broadcast network design for routing in mobile ad-hoc network. *IEEE Access* 8:188269–188283
5. Kumar V, Kalam S, Das AK, Sinha D (2021) Attack detection scheme using deep learning approach for iot. *Adv Comput Sys Sec* 14:17–30
6. Qian L, Gintautas V (2019) Fusing sensor data with publicly available information (PAI) for autonomy applications. In: *SPIE Defense + Commercial Sensing*, Baltimore, p. 22 . <https://doi.org/10.1117/12.2518933>
7. Ginoulhac R, Barbaresco F, Schneider JY, Pannier JM, Savary S (2019) Target classification based on kinematic data from AIS/ADS-B, using statistical features extraction and boosting. *Proceedings International Radar Symposium 2019-June*(681), 1–10 . <https://doi.org/10.23919/IRS.2019.8768094>
8. Qian L, Woods N, Rahman M (2019) Identifying ADS-B flight patterns. In: Paper presented at the 2019 MSS Joint (BAMS and NSSDF) Conference, San Diego, CA, 21–24 Oct 2019
9. Kraus P, Mohrdieck C, Schwenker F (2018) Ship classification based on trajectory data with machine-learning methods. In: Rohling, P.D.H. (ed.) *Proceedings International Radar Symposium*, vol. 2018-June, pp. 1–10. IEEE Computer Society, Bonn, Germany . <https://doi.org/10.23919/IRS.2018.8448028>
10. Andersson M, Johansson R (2010) Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations. 2010 International Waterside Security Conference, WSS 2010 (December) . <https://doi.org/10.1109/WSSC.2010.5730221>
11. Kumar SG, Corrado SJ, Puranik TG, Mavris DN (2021) Classification and analysis of go-arounds in commercial aviation using ads-b data. *Aerospace* 8(10):291. <https://doi.org/10.3390/aerospace8100291>
12. Mohammadi M, Al-Fuqaha A, Sorour S, Guizani M (2017) Deep learning for IoT big data and streaming analytics: a survey. *arXiv* 20(4): 2923–2960.
13. Department of the air force posture statement fiscal year 2022. department of the air force. [https://www.armed-services.senate.gov/imo/media/doc/FY22%20DAF%20Posture%20Statement%20-%20Final%20\(v23.1\)1.pdf](https://www.armed-services.senate.gov/imo/media/doc/FY22%20DAF%20Posture%20Statement%20-%20Final%20(v23.1)1.pdf)
14. Silverman BG, Bharathy G, Weyer N (2019) What is a good pattern of life model? Guidance for simulations. *Simulation* 95(8):693–706. <https://doi.org/10.1177/0037549718795040>
15. Mikhailov S, Kashevnik A, Smirnov A (2020) Tourist behaviour analysis based on digital pattern of life. In: 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), vol. 1, pp. 622–627 . <https://doi.org/10.1109/CoDIT49905.2020.9263945>
16. Hatcher WG, Yu W (2018) A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* 6:24411–24432. <https://doi.org/10.1109/ACCESS.2018.2830661>
17. Forti N, Millefiori LM, Braca P (2019) Unsupervised extraction of maritime patterns of life from automatic identification system data. In: *OCEANS 2019 - Marseille*, pp. 1–5 . <https://doi.org/10.1109/OCEANSE.2019.8867429>
18. Faraj F (2021) Object detection and pattern of life analysis from remotely piloted aircraft system acquired full motion video. PhD thesis, Queen's University , (Canada)
19. Basrawi K (2021) Aircraft classification from ads-b kinematic data. PhD thesis, Air Force Institute of Technology
20. United States department of transportation: Airspace (2021). https://www.faa.gov/air_traffic/technology/equipadsb/research/airspace/
21. National archives and records administration: electronic code of federal regulations (2021). <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-91/subpart-C/section-91.225>
22. ADS-B Exchange: home - serving the flight tracking enthusiast - ADS-B exchange (2021). <https://www.adsbexchange.com/>
23. Basrawi K, Dill R, Peterson G, Borghetti B, Lopez J (2022) Aircraft identification from ADS-B Kinematic Data. *J DoD Res Eng* 5(2):31–43

24. Sun J, Ellerbroek J, Hoekstra J (2016) Large-scale flight phase identification from ads-b data using machine learning methods. In: 7th International Conference on Research in Air Transportation, pp. 1–8
25. Sun J, Ellerbroek J, Hoekstra J (2017) Flight extraction and phase identification for large automatic dependent surveillance-broadcast datasets. *J Aerospace Inform Sys* 14(10):566–572
26. Ruseno N, Lin C-Y, Chang S-C (2022) Uas traffic management communications: the legacy of ads-b, new establishment of remote id, or leverage of ads-b-like systems? *Drones* 6(3):57. <https://doi.org/10.3390/drones6030057>
27. Filippone A, Parkes B, Bojdo N, Kelly T (2021) Prediction of aircraft engine emissions using ads-b flight data. *Aeronautical J* 125(1288):988–1012. <https://doi.org/10.1017/aer.2021.2>
28. Hasin F, Munia TH, Zumu NN, Taher KA (2021) Ads-b based air traffic management system using ethereum blockchain technology. In: 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), pp. 346–350. <https://doi.org/10.1109/ICICT4SD50815.2021.9396828>
29. Pearce N, Duncan KJ, Jonas B (2021) Signal discrimination and exploitation of ads-b transmission. In: SoutheastCon 2021, pp. 1–4. <https://doi.org/10.1109/SoutheastCon45413.2021.9401909>
30. Sun J (2021) The 1090 megahertz riddle: a guide to decoding mode s and ads-b signals. TU Delft OPEN Publishing
31. Karim F, Majumdar S, Darabi H, Chen S (2018) LSTM fully convolutional networks for time series classification. *IEEE Access* 6:1662–1669. <https://doi.org/10.1109/ACCESS.2017.2779939>
32. Karim F, Majumdar S, Darabi H, Harford S (2019) Multivariate lstm-fns for time series classification. *Neural Net* 116:237–245. <https://doi.org/10.1016/j.neunet.2019.04.014>
33. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, Cambridge
34. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
35. Alwosheel A, van Cranenburgh S, Chorus CG (2018) Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *J Choice Model* 28:167–182. <https://doi.org/10.1016/j.jocm.2018.07.002>
36. Cho J, Lee K, Shin E, Choy G, Do S (2015) How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv preprint arXiv:1511.06348*
37. Jain AK, Chandrasekaran B (1982) 39 dimensionality and sample size considerations in pattern recognition practice. In: Elsevier (ed.) Classification pattern recognition and reduction of dimensionality. handbook of statistics, vol. 2, pp. 835–855. Elsevier B.V., North Holland, Amsterdam. [https://doi.org/10.1016/S0169-7161\(82\)02042-2](https://doi.org/10.1016/S0169-7161(82)02042-2)
38. Baum EB, Haussler D (1989) What size net gives valid generalization? *Neural Comput* 1:151–160
39. Haykin SS (2009) Neural networks and learning machines. Prentice Hall, New York
40. Honeywell aerospace: automatic dependent surveillance-broadcast will transform air traffic control. technical report, honeywell international Inc, Phoenix, Arizona (2018). <https://aerospace.honeywell.com/content/dam/aerobt/en/documents/landing-pages/white-papers/DO-260-mandates-white-paper.pdf>
41. Kamaruzzaman MRB, Fall D, Hossain MD, Taenaka Y, Kadobayashi Y (2022) Modulated synchronous taxiing: Mitigating uncertainties amid ads-b spoofing. In: 2022 Integrated Communication, Navigation and Surveillance Conference (ICNS), pp. 1–13. <https://doi.org/10.1109/ICNS54818.2022.9771477>
42. Riahi Manesh M, Kaabouch N (2017) Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ads-b) system. *Int J Crit Infrastruct Prot* 19:16–31. <https://doi.org/10.1016/j.ijcip.2017.10.002>
43. Wu Z, Shang T, Guo A (2020) Security issues in automatic dependent surveillance - broadcast (ads-b): a survey. *IEEE Access* 8:122147–122167. <https://doi.org/10.1109/ACCESS.2020.3007182>
44. Kaur H, Pannu HS, Malhi AK (2019) A systematic review on imbalanced data challenges in machine learning: applications and solutions. *ACM Comput Surv (CSUR)* 52(4):1–36
45. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: review of methods and applications. *Expert Syst Appl* 73:220–239
46. Weinman JJ, Lidaka A, Aggarwal S (2011) TensorFlow: Large-scale machine learning. GPU Computing Gems Emerald Edition (November), 277–291. <https://arxiv.org/abs/1603.04467v2>
47. Brooks TN (2018) Using autoencoders to learn interesting features for detecting surveillance aircraft. <https://doi.org/10.48550/ARXIV.1809.10333>

48. Haygood J (2021) Turboprop Vs piston airplanes . <https://www.skytough.com/post/turboprop-vs-piston-airplanes>
49. Service NW (2022) Pressure definitions . https://www.weather.gov/bou/pressure_definitions/

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Sarah Bolton¹  · Richard Dill¹ · Michael R. Grimaila² · Douglas Hodson¹

Richard Dill
richard.dill@afit.edu

Michael R. Grimaila
michael.grimaila@afit.edu

Douglas Hodson
douglas.hodson@afit.edu

¹ Department of Engineering and Computer Science, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB 45433, OH, USA

² Department of Systems Engineering and Management, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB 45433, OH, USA