AIAA AVIATION Forum
August 2-6, 2021, VIRTUAL EVENT
AIAA AVIATION 2021 FORUM

10.2514/6.2021-2405

Deep Autoencoder for Anomaly Detection in Terminal Airspace Operations

Samantha J. Corrado*, Tejas G. Puranik†, Olivia Pinon Fischer‡, Dimitri N. Mavris§
Aerospace Systems Design Laboratory, Georgia Institute of Technology, Atlanta, GA, 30332, U.S.A.

Rodrigo L. Rose¶
Georgia Institute of Technology, Atlanta, GA, 30332, U.S.A.

Roohollah Heidary‖, Jesse Williams**
Global Technology Connection, Atlanta, GA, 30339, U.S.A.

In recent years, the aviation industry has seen a large increase in the volume of operations. The maintenance and improvement of safety at acceptable levels is one of the most important concerns in civil aviation operations. Reactive methods to aviation safety improvement are being augmented with proactive and predictive approaches that leverage large amounts of routinely collected aviation data. Due to the increased availability of airborne sensor data and improvements in computing power, application of machine learning methods to various aviation safety problems for identifying, isolating, and reducing risk has gained momentum. Previous work in this domain has focused on identifying anomalies or abnormal operations as a first step towards identification of potentially risky situations using aircraft sensor data. However, most existing methods rely only on the aircraft data and do not take into consideration the environment and context in which it is operating. In this paper, a novel framework based on deep learning methods using autoencoders is proposed to identify anomalies in terminal airspace operations. Data from multiple sources (aircraft trajectory, weather, traffic/congestion) is fused and utilized in the model development process which has not been attempted in prior work. The framework is proposed with the central idea of using historical aircraft trajectory data fused with weather and traffic metrics to build an anomaly detection model to identify trajectories that deviation from the norm, given a specific context. The framework is demonstrated on six months of arriving flight data collected for San Francisco International Airport as a case study. The developed framework has the potential to aid air traffic controllers in identifying high risk situations from a holistic perspective and applying appropriate mitigation strategies.

I. Introduction

The transformation of aviation systems is occurring at a faster rate than ever before as technologies developed within various aviation disciplines continue to evolve and coalesce. In recent years, the aviation industry has seen a large increase in the volume of operations. Aviation demand is driven by economic activity, where the growing U.S. and world economies provide the foundation for long term aviation growth. According to the Federal Aviation Administration (FAA), the demand for air travel and traffic is predicted to grow steadily over the next two decades at a rate of 1.8% over the next 20 years, making efficient and safe operations more important than ever [1]. Along with the volume of operations, the complexity of air operations is also expected to increase with the introduction of Unmanned Aircraft Systems (UAS) traffic management (UTM), urban air mobility (UAM) in the National Airspace System (NAS). With such a large increase in the volume and complexity of expected operations, there is an ever-increasing demand

*Senior Graduate Researcher, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, AIAA Student Member
†Research Engineer II, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, AIAA Senior Member
‡Senior Research Engineer, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, AIAA Senior Member
§S.P. Langley NIA Distinguished Regents Professor and Director of Aerospace Systems Design Laboratory, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, AIAA Fellow.
¶Undergraduate Student, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology
‖Senior R&D Engineer, Global Technology Connection
**Chief Technology Officer, Global Technology Connection

for maintaining or improving safety for all aviation operations. In response to air traffic growth, global efforts have been underway to modernize aviation systems to address current and future air transportation challenges. These global modernization efforts include the FAA's Next Generation Air Transportation System (NextGen) [2] portfolio in the U.S. and the Single European Sky Air traffic management (ATM) Research (SESAR) [3] program in Europe. All global modernization efforts are long-term plans motivated by increasing efficiency and capacity of airspace systems, while also maintaining safety.

In the context of current operations, flight trajectory is the core information that is used by the air traffic management (ATM) system as a basis for distributing flight information to relevant airlines and air traffic control (ATC) units, facilitating timely coordination between sectors and units, correlating flight data with tracks, monitoring the adherence of an aircraft with its assigned route, and detecting and resolving conflicts The identification of significant events in historical aircraft trajectory data falls under the scope of knowledge discovery and information extraction. A category of applications which is the most popular in the aviation safety domain is anomaly detection using quantitative time-series data and semi-supervised or unsupervised machine learning techniques. In such applications, historical data recorded from routine operations is analyzed using machine learning techniques to identify flights or trajectories that deviate from nominal operations. Anomaly detection is an important step in safety improvement of the air transportation system as evidenced by the numerous applications in recent years [4–8]. However, an inherent limitation of existing approaches is that the anomalies are identified only with respect to the aircraft dynamics, trajectory, etc., and do not consider external factors such as the environmental conditions and/or the state of the system (traffic, congestion, etc.) at the time of operation. This exerts a limitation on analysis of the identified anomalies, specifically when trying to isolate causal factors.

As illustrated in Figure 1, anomalies could result from weather conditions, ATC actions such as deconfliction and sequencing, aircraft dynamics, or any combination of these factors. For example, weather events can have a significant impact on airport performance and cause delayed operations if the airport capacity is constrained. Adverse weather encounters during flight can affect the trajectory and energy state of the aircraft, thereby causing unusual/anomalous behavior. Identifying the root cause of trajectory anomalies in the air transportation system can be difficult without accurate weather information, and system-level metrics related to congestion, traffic, etc., especially during adverse weather event conditions.

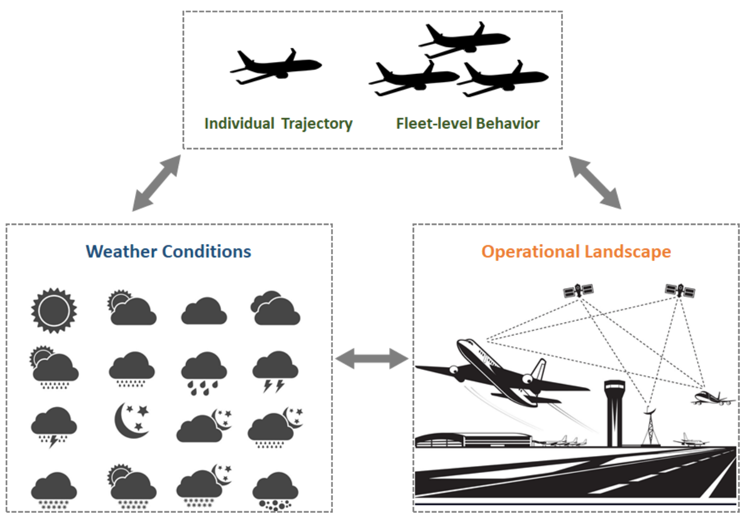

Fig. 1 Various types of data that can be used for anomaly detection in terminal airspace operations

Despite numerous implementations of anomaly detection using flight data, there are limited frameworks that enable using fused data from multiple sources for detecting anomalies. An inherent limitation of such approaches is that the anomalies are identified only with respect to the aircraft dynamics, trajectory, etc., and do not consider external factors such as the environmental conditions and/or the state of the system (traffic, congestion, etc.) at the time of operation. Fig. 1 represents the different contexts that could be considered when detecting anomalies and precursors to anomalies in trajectories. The fusion of multi-source data is thus necessary to ensure that the underlying uncertainties are captured and potential causes discovered. Therefore, a framework that explicitly accounts for these factors (weather information, system-level metrics related to congestion, traffic, etc.) for the purpose of anomaly detection is proposed. Considering the previous observations, the following research objective is proposed for this work:

Research Objective: The overarching research objective of this effort is to develop a flexible, state-of-the-art framework utilizing historical trajectory data fused with weather and traffic metrics to build an anomaly detection model to identify trajectories that deviate from the norm, given a specific context.

II. Background

Prior to 1995, aviation safety was typically *reactive*, where after an accident or incident occurs, then a mitigation strategy was developed and implemented [9]. A reactive approach requires an accident or incident to be experienced, where, subsequently, the underlying problem is identified and addressed. A reactive approach is unable to keep safety at adequate levels as air traffic increases and aviation systems are modernized. Thus, the aviation industry realized new approaches must be developed [9].

In the past 25 years, the industry has made efforts to shift toward *proactive* and *predictive* approaches to safety. A proactive approach to safety involves identifying potential unsafe events before they manifest as accidents or incidents such that mitigation strategies may be developed to prevent the occurrence of accidents or incidents related to the unsafe events [10]. Taking safety analysis a step further, a *predictive* approach to safety involves monitoring data obtained from routine operations in addition to accidents and incident data and reports to detect potential negative future outcomes. A reactive approach to safety focuses on prevention of accident or incident *recurrence*, while proactive and predictive approaches to safety focus on prevention of accident or incident *occurrence*. The shift to proactive and predictive safety is key to maintaining aviation safety in the future as aviation systems modernize and becomes more complex.

To support proactive and predictive approaches to safety, enabling robust anomaly detection is paramount. Anomaly detection may be defined as *the process of detecting rare instances or sets of instances in a data set that may be of concern due to behaviors or characteristics that differ from the rest of the data set*. Aviation anomaly detection method development has focused on leveraging data-driven, machine learning approaches. There exist *supervised*, *semi-supervised*, and *unsupervised* machine learning algorithms that may be leveraged for anomaly detection. The key distinction between supervised and unsupervised machine learning methods is the presence of *labels* within the data set. As aviation data is typically *unlabeled* with respect to anomalies, i.e. there does not typically exist a parameter in aviation data sets that explicitly indicates whether a data instance is an anomaly or not. Accordingly, the aviation anomaly detection task is often formulated as an unsupervised anomaly detection problem.

Several unsupervised anomaly detection methods exist with aviation literature. Recently, Basora & Olive published an up-to-date review on the recent advances in anomaly detection methods applied to aviation data [11]. As there is a vast array of anomaly detection methods that have been developed, only the most relevant and prevalent will be discussed. One of the first and most effective anomaly detection methods is multiple kernel anomaly detection (MKAD), developed by Das et al. [4], which detects anomalies in heterogeneous sequences of both continuous and discrete features. Li et al. [12] present ClusterAD, a clustering-based anomaly detection method leveraging the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm. Li et al. [5] extend this work to develop a ClusterAD-Flight, which identities anomalies at the flight level. Additionally, Li et al. [13] present ClusterAD-DataSample, which leverages a Gaussian Mixture Model (GMM) to detect instantaneous anomalies. Both Das et al. [4] and Li et al. [5, 12, 13] focus on detecting anomalies provided Flight Operational Quality Assurance (FOQA) data. FOQA data is collected as part of a structured routine data collection and analysis program, where Flight Data Recorders (FDR) on-board aircraft record massive amount of data for each flight. The amount of operational data recorded includes anywhere between 80 to 2,000 metrics at a sampling rate of 0.25 to 8 Hz [14]. Access to FOQA data is often very limited to individual airlines and aviation operators.

As a product of aviation system transformation efforts, new data-generating technologies have been deployed in aviation systems, which has resulted in more sources, volume, and availability of aviation operational data than ever before [2]. For instance, Automatic Dependent Surveillance-Broadcast (ADS-B) technology has been deployed to enhance aviation safety and efficiency by enabling aircraft to determine their position with respect to other similarly-equipped aircraft, using satellite, inertial, and radio navigation [15]. ADS-B Out periodically emits (at approximately 1 Hz) the aircraft's position, along with other relevant parameters, to ground stations and other equipped aircraft [15]. The expansion of ADS-B technology has enabled open-source flight trajectory data to become publicly available. Detecting anomalies in aircraft trajectory data, such as that provided by ADS-B technology, has become an active area of research.

To detect anomalies in general aviation trajectory data, Puranik et al. [16] define energy metrics to be computed from trajectory data records and utilized for anomaly detection. Puranik & Mavris [7] present a method leveraging DBSCAN and SVM for detecting anomalies in departing and arriving general aviation flights. Focusing on detection of anomalies in commercial aviation data, Kim & Hwang [17] and Deshmukh [18] propose TempAD, an algorithm designed to provide formulas related to the bounds of normality that are easily interpreted in natural languages, where this method utilizes DBSCAN to identify air traffic flows as a data pre-processing step. As identification of air traffic flows is a typical pre-processing step when detecting anomalies in trajectory energy-related metrics, Corrado et al. [19] present a method to more accurately identify air traffic flows using a weighted Euclidean distance function. Jarry et al. [20] leverage functional principal components analysis (FPCA) and HDBSCAN (Hierarchical DBSCAN) with GLOSH (Global-Local Outlier Score from Hierarchies) to detect anomalies within the terminal airspace. Finally, Olive & Basora [21] present an autoencoder-based framework to detect anomalies, or significant operational events, in trajectory data. However, Olive & Basora consider only true track angle as the feature utilized to indicate an anomalous trajectory [21]. Corrado et al. [22] recently presented a distinction between anomalies identified in the spatial and energy dimensions of ADS-B trajectory data and associated derived metrics, where both HDBSCAN and DBSCAN were leveraged to perform the anomaly detection. Further, a primary gap is identified in that none of the aforementioned methods have leveraged additional data sources in the anomaly detection stage. Rather, additional data has been considered only after the fact, to aid in validation.

III. Method

The central idea of this framework is to use historical trajectory data fused with weather and traffic data to build an anomaly detection model to identify trajectories that deviate from the norm, given a specific context. In the proposed work, the definition of a trajectory anomaly is adapted using the definition of anomalies from Chandola et al. [23] as "a significant deviation from the norm for a particular flight with respect to its trajectory parameters such as altitude, speed, descent/climb rate, and flight track (latitude and longitude)". The parameters that define the trajectory in this context are closely tied to the kinematic energy state of the aircraft (kinetic, potential, total mechanical energy) and are thus closely related to energy-based safety metrics defined in our previous work [16].

While there are multiple options for anomaly detection algorithms, deep learning methods are particularly suited for learning complex relationships from large amounts of raw data without the need for manually designing features. In doing so, they circumvent the significant amount of domain expertise and time needed to handcraft features. They have also shown to address the over-fitting and lack of generalizability common to traditional machine learning techniques [24]. In particular, autoencoders are a category of neural networks capable of learning non-linear reduction functions as well as their inverse functions to transform data into a low dimensional space (called encoded or latent space) and back into the original representation. The reconstruction error from the autoencoder is used as a method of identifying trajectory anomalies in the proposed framework. A discussion on the application of autoencoders for anomaly detection is presented below. The overall framework proposed in this work is shown in Figure 2. Each block of the proposed framework is discussed in greater detail below.

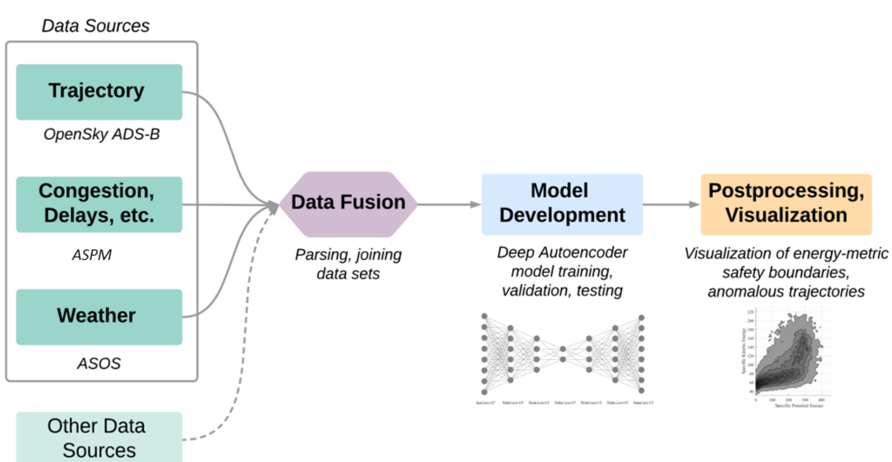

Fig. 2 Overall anomaly detection framework

A. Data Sources

Considering the overarching research objective of this effort, multiple data sources are required to build an anomaly detection model to identify trajectories that deviate from the norm, given a specific context. Specifically, these data sources may include trajectory data, such as that provided by ADS-B technology or FOQA data, as introduced, weather data, and traffic data, which includes information related to congestion and delays within the airspace system.

It is noted here that the design of the anomaly detection framework is meant to provide flexibility to include other data sources as required/possible. Additionally, the design of the anomaly detection framework is meant to be versatile related to the fidelity of the data, i.e. high quality trajectory data such as FOQA data may be included in the data sources in this framework, or lower-quality ADS-B data may be sufficient, dependent upon the application. Similarly, weather data sources are of vary accuracy and frequency. Furthermore, it is possible that multiple sources for one type of data may be used and fused according to their relative errors to provide a more accurate measurement. Regardless of the specific source or fidelity of data leveraged, data from a specific geographic location of interest over a period of several months from multiple airlines is necessary to demonstrate the value of this framework. This may translate into over a hundred thousand flights, weather readings, and traffic/congestion data during that time period.

B. Data Fusion

Prior to fusing all selected data sources, the extracted data from each source must be cleaned in such a way that enables easy fusion. Thus, the data fusion block includes any necessary data extraction, cleaning, fusion, processing, and augmentation required to generate a fused feature vector for each flight to be input into the machine learning model. Data fusion in this context is generally time-based, such that weather and traffic metrics are associated to a trajectory based on aligning and interpolating metric measurements at various times. In the case that multiple data sources are available to provide the same information, these data sources may be fused according to their relative errors leveraging techniques such as Kalman filtering [25].

C. Model Development

As introduced, deep learning methods are well-suited for anomaly detection, particularly the use of autoencoders for unsupervised anomaly detection. Thus, this framework proposes the development of an autoencoder model for anomaly detection provided the generated feature vectors. As mentioned in Section II, autoencoders have previously been utilized within the aviation literature to detect anomalies. However, these methods have not made use of additional data sources. This section first provides a detailed explanation of the concept of autoencoders. Then, an overview of utilization of autoencoders for anomaly detection within the aviation literature is presented.

1. Autoencoder Models

An *autoencoder* is an unsupervised deep learning algorithm defined as “the combination of an encoder function, which converts the input data into a different representation, and a decoder function, which converts the new representation back into the original format” [26]. Autoencoders reduce dimensionality by setting the number of extracted features to be less than the number of inputs, and they are usually trained by backpropagation in an unsupervised manner. The underlying optimization problem aims to minimize the distance between the reconstructed results and the original inputs, i.e. the *reconstruction error*. Depending on the volume and complexity of the data set considered, autoencoders’ layers may be stacked (multiple hidden layers) in an attempt to better capture the structure of the underlying trajectories. The autoencoder’s architectural complexity in terms of the number of layers and the number of nodes in each layer influences the models fitting performance through the complexity of features defined in the encoding/decoding layers. An increase in the layers and nodes results in more sophisticated features embedded in the encoder and decoder. This is apparent when observing the sharp rise in the number of independent parameters when increasing the model complexity.

A sample of an autoencoder model architecture in the context of the proposed framework is presented in Figure 3. As per the objective of this work, weather data, traffic data, and trajectory data are combined to produce feature vectors which are inputs into the autoencoder model, i.e. the input layer. Then, hidden layers are added (which include the latent space layer) in a symmetric fashion around the latent space layer to finally build out to the output layer, which are the resulting reconstructed trajectories. The layers to the left, between the input layer and the latent space layer, perform the task of *encoding* the input data. The layers to the right, between the latent space layer and the output layer, perform the task of *decoding* the latent space layer to produce a reconstruction of the input data. It is noted that before inputting the data into the autoencoder model, feature vectors are often scaled using scaling techniques such as z-normalization or min-max scaling. Normalization is standard when applying machine learning algorithms to reduce bias towards features with large magnitudes during the training process.

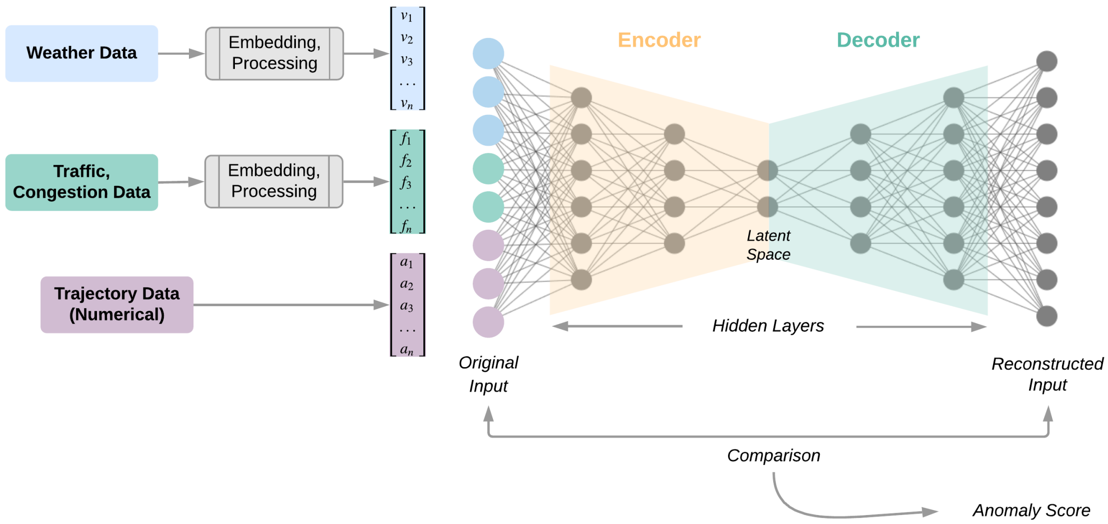

Fig. 3 Notional autoencoder architecture

The key to leveraging autoencoders for anomaly detection lies in assessment of the reconstruction error, which is computed with respect to some defined *loss function*, such as the mean squared error (MSE). As depicted in Figure 3, it is typical within autoencoder anomaly detection literature to take the reconstruction error as an *anomaly score*. It is assumed that anomalies are rare events, making up only a very small portion of the data set. If the autoencoder is not overfit, then the autoencoder should learn the intrinsic structure of the nominal data set, as it makes up the majority of the data set. On the other hand, the rare behavior of the anomalies should not be learned. Therefore, the model should more successfully reconstruct nominal trajectories as opposed to the anomalous trajectories. It is possible to take the trajectories with reconstruction errors, or anomaly scores, above a certain threshold as the detected anomalies. This threshold may be derived from desired percentiles or a set value.

2. Autoencoders for Anomaly Detection in the Aviation Literature

In recent years, there have been a few studies utilizing autoencoders for anomaly detection in trajectory data, specifically utilizing OpenSky Network data. Olive et al. [27] present a framework to analyze flight trajectories, detect unusual behavior, and infer ATC actions. This work analyzes only en-route trajectory data between city-pairs [27]. Further, only the true track angle, or heading, feature is utilized as input into the autoencoder model [27]. When an anomaly is detected utilizing this approach, there is an attempt to place the anomaly in context to infer whether the anomaly would correspond to an ATC action [27]. Olive & Basora [28] expand on this work to attempt to identify operationally significant events which can be associated with ATC actions in all en-route traffic within the LFBBPT sector in the French Bordeaux Area Control Center. Again, only the true track angle feature is utilized as input to the autoencoder [28]. Most recently, Olive & Basora [29] again expand this work to present an autoencoder-based framework to detect anomalies, or significant operational events, in trajectory data including city-pair, en-route, and terminal airspace trajectory data. However, Olive & Basora, again, consider only true track angle as the feature utilized to indicate an anomalous trajectory [29], where this is an identified limitation of the previously-mentioned methods. Thus, a primary gap consistent with that which motivated development of the proposed framework is that none of the aforementioned methods have leveraged additional data sources in the anomaly detection stage. Rather, additional data has been considered only after the fact, to aid in validation. Thus, the model development is tailored to address this gap through inclusion of additional features.

D. Post-Processing and Visualization

The final block of the proposed framework involves post-processing and visualizing results obtained from the model. Raw model results are not especially useful to operators nor decision-makers. Thus, these results must be presented in such a way that actionable insights may be obtained. Further, visualization of model results may additionally serve as a method of qualitative final model validation. For instance, it is important to visualize flight trajectories both individually and in the context of the entire data set, especially those trajectories which may have been detected as being anomalous. Specific visualizations required are use-case dependent.

IV. Implementation and Results

An implementation of the framework introduced in Section III is presented. The San Franscisco International Airport (KSFO) terminal airspace was selected to perform the implementation. Specifically, flights arriving at KSFO from January 1^{st}, 2019 through June 30^{th}, 2019 were considered. Thus, trajectory, weather, and traffic data were extracted for flights arriving at KSFO during this time period. The data sources leveraged for implementation of the proposed methodology are discussed. Subsequently, a discussion on the data fusion framework block is presented, followed by a discussion on the model development. Finally, the post-processing and visualization of the raw model results are discussed and presented.

A. Data Sources

Three primary data sources are considered in this effort: (1) OpenSky Network data (trajectory data), (2) Aviation System Performance Metrics (ASPM) data (traffic data), and (3)Automated Surface Observing System (ASOS) data (weather data). A brief introduction to each is presented.

1. OpenSky Network

To identify trajectory anomalies using energy-based metrics, a source of ADS-B based aircraft position, velocity, and status data is required. The OpenSky Network [30] is a non-profit association that processes and archives ADS-B data from a global network of sensors. OpenSky data has previously been used by researchers for a diverse range of studies. The *traffic* library, enables downloading OpenSky Network historical ADS-B trajectory data, where each data record is referred to as a *state vector*. State vectors contain timestamps (added on the receiver side, with many receivers equipped with a GPS nanosecond precision clock), transponder unique 24-bit identifiers (icao24), space-filled 8 character callsigns, latitude, longitude (in degrees), (barometric) altitude (in feet, w.r.t. standard atmosphere), GPS altitude (in feet), ground speeds (in knots), true track angle (in degrees), vertical speed (in knots). Table 1 provides an example of the structure and contents of two state vectors obtained from the OpenSky Network for an aircraft.

epoch time	icao24	callsign	heading	baroaltitude	lat	lon	...
1546300800	1	3	48.628	70.52	53.6326	9.992	...
1546300801	1	3	49.161	268.66	53.6392	10.004	...
1546300802	1	3	50.240	457.43	53.6459	10.018	...

Table 1 Example structure and contents of data obtained from the OpenSky Network historical database

A decent ADS-B receiver antenna can receive messages from cruising aircraft located up to 250 miles away. The range is typically lower for aircraft flying at lower altitudes. The trajectory data available from the OpenSky Network can be leveraged to evaluate energy-based metrics mentioned earlier using the aircraft's position, velocity, altitude, etc. The OpenSky Network has good coverage over Europe and North America and any airport under the coverage can be selected for analysis.

2. Aviation System Performance Metrics (ASPM)

The ASPM database* is one of the FAA's four core databases that are utilized to produce the operational metrics that are tracked and reported to manage FAA efficiency. The ASPM database derives its metrics utilizing information from the Traffic Flow Management System (TFMS). TFMS is used by air traffic management personnel to plan and execute traffic flow management initiatives to ensure that constrained areas in the National Airspace System remain safe and operate optimally. TFMS is comprised of two message streams: TFMS Flight and TFMS Flow. The TFMS Flight message stream provides initial flight plan messages, amended flight plan messages, departure and arrival time

*Source: https://aspm.faa.gov/

notifications, flight cancellation messages, boundary-crossing messages, and track position reports. The TFMS Flow message stream, on the other hand, provides data on traffic flow management initiatives such as Ground Stops, Reroutes, Airspace Flow Programs etc. Metrics such as airport arrival and departure rates, delay rates for various categories of delays, and on-time percentages are available from the ASPM database on an hourly basis. Extracting these metrics from the ASPM database enables system-level traffic information to be provided as context to the anomaly detection algorithm. A typical example of processed ASPM data is shown in Table 2. This data can provide further insights into trajectory anomalies that might not otherwise be available.

hour	date	scheduled departures	% on-time departures	...
0	01/01/2019	4	75	...
1	01/01/2019	1	100	...
2	01/01/2019	6	85	...

Table 2 Example structure of data obtained from ASPM database

3. Automated Surface Observing System (ASOS)

The ASOS program is a joint effort of the National Weather Service (NWS), the FAA, and the Department of Defense (DoD). ASOS units are automated sensor suites that are designed to serve meteorological and aviation observing needs and are widely used by meteorologists, climatologists, hydrologists, and aviation weather experts. The ASOS systems serve as the nation's primary surface weather observing network. ASOS works non-stop, updating observations every minute, 24 hours a day, every day of the year. A basic strength of ASOS is that critical aviation weather parameters are measured where they are needed most: airport runway touchdown zone(s). There are currently more than 900 ASOS sites in the United States[†]. For this work, the Iowa Environmental Mesonet[‡] is used to obtain ASOS data. An example of the historical ASOS data available from the Iowa Environmental Mesonet is shown in Table 3.

time	tmpf	dwpf	relh	...
01/01/2019 00:56	54.0	30.9	41.11	...
01/01/2019 01:05	54.0	32.0	42.98	...
01/01/2019 01:09	54.0	31.5	43.00	...

Table 3 Example structure data obtained from the IEM database

B. Data Fusion

As mentioned, the data fusion block of the framework includes the data extraction, cleaning, fusion, processing, and augmentation required to generate a fused feature vector or each trajectory. The data extraction, cleaning and fusion procedure contained three primary steps: (1) Initial Cleaning, (2) Segment Identification, and (3) Final Cleaning. The data fusion steps are presented in Figure 4. The OpenSky Network trajectory data was first extracted and was initially cleaned in the first step of the data extraction, cleaning, and fusion procedure. Then, using only the OpenSky Network data, flight segment identification occurred. Next, the ASOS weather data was fused and leveraged in the final cleaning step, after which the ASPM traffic data was fused. It is noted that, in this implementation, the trajectory data metrics are time series data, while the weather and traffic data metrics are metadata, meaning a single value for each metric is associated with the entire trajectory, i.e. a trajectory may have n time series data points for each metric, yet only one data point for each weather and traffic data metric. Each of the data extraction, cleaning, and fusion steps are discussed in more detail below as well as the data processing and data augmentation steps.

[†]Source: https://www.weather.gov/asos/asostech
[‡]Source: https://mesonet.agron.iastate.edu/request/download.phtml?network=CA_ASOS

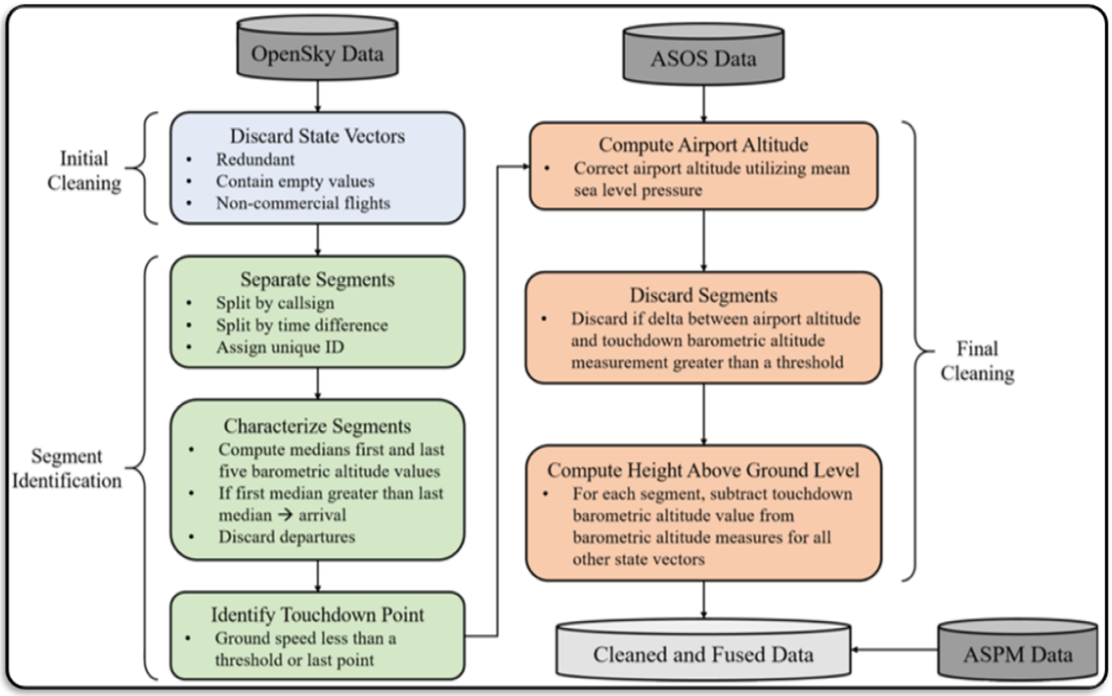

Fig. 4 Data fusion procedure

1. Initial Cleaning

OpenSky Network state vectors are extracted from OpenSky Network's historical database leveraging the *traffic* [31] Python library. The raw extracted state vectors from the OpenSky Network require a few steps to initially clean the data. These steps proceed as:

1) Remove state vectors with redundant values for latitude, longitude, barometric altitude, ground speed, and heading measurements.
2) Remove state vectors with empty values or latitude, longitude, barometric altitude, ground speed, and heading measurements.
3) Remove state vectors not corresponding to a commercial flight operation, determined by parsing the callsign associated with the state vector.

2. Segment Identification

Next, trajectory segment identification occurred. The state vectors were split by callsign, and then further split into individual trajectory segments where the time difference between two successive state vectors was greater than five minutes. Five minutes was set as the threshold for which to separate trajectory segments due to the loss of information that would result from five minutes between trajectory measurements within the terminal airspace. Further, it is possible that multiple flights operate with the same callsign in one day, and splitting by a large time difference enables both of these segments to be captured. A unique identifier was assigned to each of the identified trajectory segments. The trajectory segments were then characterized as arrival or departure segments based on the medians of the first and last five altitude measurements. If the first altitude median was greater than the last altitude median, then the trajectory segment was characterized as an arrival. Otherwise, the trajectory segment was characterized as a departure and disregarded for the purpose of this work. The median of the first five altitude measures was taken to be robust to any noise or erroneous measurements that occur.

Then, the touchdown point state vector was identified. The touchdown point was identified as the first state vector where the ground speed reaches below 100 knots, or the last point if no point was recorded to reach below 100 knots. It is noted that this is an approximate method for identifying the touchdown point and can lead to some error, but is observed to be the most robust for this data set.

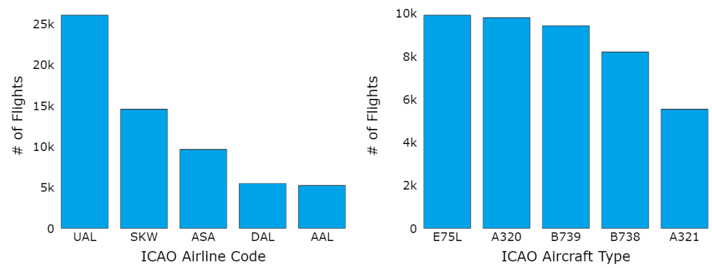

Fig. 5 Top five most common airlines and aircraft types

Before final data cleaning steps, the ASOS data was fused with the OpenSky Network data, which first required some cleaning of the raw extracted ASOS data. The raw extracted ASOS data from the Iowa Environmental Mesonet required filling missing values in the columns 'tpmf', 'dwpf', 'relh', 'drct', 'sknt', 'p01i', 'alti', 'mslp' and 'gust' by performing nearest-neighbor interpolation. The ASOS data was associated with each flight segment based on the touchdown vector's timestamp.

3. Final Cleaning

Finally, the final cleaning of data proceeded. The lowest trajectory segment altitude reached was determined by assessing the mean sea level pressure from the ASOS data and computing a corrected airport altitude for the touchdown point state vector. The trajectory segment was discarded if the difference between the touchdown altitude and the airport altitude was greater than 100 ft (i.e. the trajectory segment did not reach within 100 ft above ground level). These segments were discarded because enough data does not exist to consider the approach to be complete. The touchdown altitude was then utilized to compute a Height Above Ground Level (HAGL) for each state vector by subtracting the touchdown altitude from all other altitude measurements. Finally, the hourly ASPM data was fused with each trajectory segment based upon the hour of the touchdown state vector's timestamp. The raw ASPM data did not require extensive data cleaning efforts.

The resulting set of ADS-B trajectory data for flights operating within the KSFO terminal airspace from January 1^{st}, 2019 through June 30^{th}, 2019 contains 81,776 total arrival trajectory segments. The data set contains both domestic and international flights on 135 airlines, where approximately one-fourth of the flights are operated by United Airlines. Further, the data set contains 88 unique aircraft types, where the most commonly operated are E75L aircraft, closely followed by A320 aircraft and B737-900 aircraft. A distribution of the top five most common airlines and aircraft types for the 81,776 flights in this data set is displayed in Figure 5.

4. Data Processing

Anomaly detection is typically performed considering a data set that has been cut off at some distance- or time-based threshold, i.e. 15 nautical miles cumulative ground track distance from the touchdown point or the final 10 minutes of flight. In this work, a 15 nautical mile distance-based cutoff was selected to mitigate the effects of potentially differing approach speeds for different aircraft type. Any trajectory points outside of this 15 nautical mile cumulative ground track distance threshold were discarded. However, this results in trajectory records of varying lengths between the different trajectories.

Anomaly detection methods typically operate on trajectory records of a standardized n-dimensional length. Therefore, a re-sampling of the time series trajectory data to form n-dimensional trajectory records was necessary. There exist two primary re-sampling methods: distance-based and time-based. In the case where aircraft maintain a relatively constant velocity, distance-based and time-based re-sampling methods generate a nearly identical set of points. However, within the terminal airspace, aircraft velocity is rarely constant for extended periods of time, particularly during the approach phase where it gradually decreases. A time-based re-sampling would result in an uneven balance of points closer to the airport. Therefore, a distance-based re-sampling method is selected. A uniform re-sampling occurs based on the cumulative ground track distance of an aircraft from its touchdown point. This results in the re-sampled points being spatially more evenly distributed. Therefore, the data set was re-sampled to provide 50 evenly-spaced points between 15 nautical miles cumulative ground track distance from the touchdown point and the touchdown point. This resulted in a resolution of about 0.3 nautical miles cumulative ground track distance between trajectory points within a trajectory segment.

5. Data Augmentation

To support anomaly detection in the energy dimension of the trajectories, the time series trajectory data was augmented with derived energy metrics. Energy metrics were computed that indicate the energy state of the aircraft at each trajectory point. The energy metrics computed to augment the data set include:

- **Specific Potential Energy**: $SPE = h$, where h is the HAGL.
- **Specific Kinetic Energy**: $SKE = \frac{V^2}{2g}$, where V is the ground speed and g is the gravity constant.
- **Specific Total Energy**: $STE = SPE + SKE$, where STE is the sum of the specific potential and specific kinetic energies.
- **Specific Potential Energy Rate**: $SPER = \frac{SPE_{i+1} - SPE_i}{\Delta t}$, where Δt is the time difference between two consecutive records, $i + 1$ and i.
- **Specific Kinetic Energy Rate**: $SKER = \frac{SKE_{i+1} - SKE_i}{\Delta t}$.
- **Specific Total Energy Rate**: $STER = \frac{STE_{i+1} - STE_i}{\Delta t}$.

The data fusion steps produce a data set containing time series metrics as well as metadata metrics. The time series metrics of interest included the latitude, longitude, ground speed, heading, vertical rate, and HAGL as well as the derived energy metrics. The metadata metrics of interest included those associated with the ASOS data set such as temperature, dew point temperature, relatively humidity, wind direction, wind speed, and one-hour precipitation and also those associated with the ASPM data set such as number of departures, number of arrivals, percent of on time gate departures, percent of on time airport departures, percent of on time gate arrivals, average gate departure delay, average taxi-out time, average taxi-out delay, average airport departure delay, average airborne delay, average taxi-in delay, average block delay, and average gate arrival delay.

To generate a feature vector for each trajectory, the metadata features were aligned in a row and the time series features were stacked and aligned in the same row. The feature vectors for each trajectory were then aggregated to produce the final feature vector matrix. An example of the structure of the feature vector matrix is displayed in Table 4. Overall, there are 23 metadata columns and 550 times series data columns (11 time series features, re-sampled to 50 points each), resulting in 573 total columns within the feature vector matrix.

unique_id	**tmpf**	**...**	**#_dep**	**...**	**lat_0**	**...**	**lat_49**	**...**
KSFO_arr_0	56	...	456	...	36.6231	...	36.8426	...
...	...	...	...	...	...	...	...	...
KSFO_arr_81776	64	...	562	...	36.6232	...	36.5426	...

Table 4 Example structure of the feature vector matrix

C. Model Development

To build the autoencoder model, the *tensorflow* [32] Python library was leveraged. As mentioned, the architecture of the autoencoder is determined by the number of layers and the size of the layers (number of nodes within the layer). Additional hyperparameters are available as input to the autoencoder model within the *tensorflow* library, such as number of training epochs ($epochs$) and batch size ($batch\_size$), which is the number of training examples per gradient update. Further, the autoencoder model is trained such that a *loss function* is minimized.

The autoencoder model architecture implemented included the following layers of the following sizes: (0) input layer of 573 nodes, (1) hidden layer of 500 nodes, (2) latent space layer of 10 nodes, (3) hidden layer of 500 nodes, and (4) output layer of 573 nodes. The *tensorflow* [32] Python library was leveraged to build the autoencoder model. The autoencoder model was trained for 50 epochs with a batch size of 150. Due to the uniqueness of the data set in that it contains both metadata and time series data features, the selection of a loss function is challenging. It is often desirable to weight all features equally when computing the loss, or reconstruction error so as to not bias results towards one feature. However, due to the presence of both metadata and time series data features, all features will not be weighted equally utilizing a standard loss function such as MSE. This is because the time series data features essentially hold about 50 times more weight, i.e. an anomaly in the velocity dimension would result in at least one, yet likely more, points within the velocity columns having abnormal values, which would increase the MSE significantly for however many points are abnormal, whereas an anomaly in a weather dimension would only contain one abnormal value and would not be capable of significantly impacting the MSE when compared to the anomaly in the velocity dimension. Consequently, a standard loss function such as the MSE is not appropriate for this application.

Therefore, a custom loss function is proposed that is a modified MSE function. In this custom function, the MSE of the individual points of time series data features are averaged for each of the time series data features' dimensions before contributing to the MSE as a whole. Figure 6 displays the intuition behind the modified MSE custom loss function that reduces bias toward time series data.

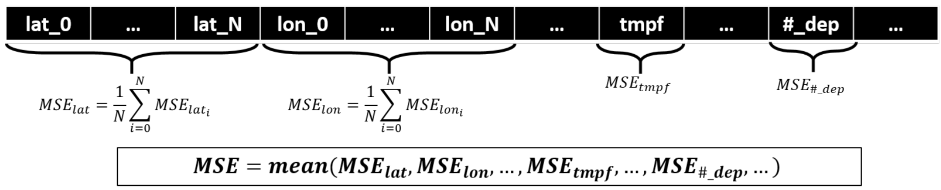

Fig. 6 Custom loss function illustrated

Before inputting the data into the autoencoder model, the feature vectors were normalized utilizing the Z-normalization technique the *StandardScaler* module from the *sklearn* [33] Python package. Z-normalization re-scales the data within each column to have zero mean and unit standard deviation: $Z = \frac{x-\mu}{\sigma}$, where Z is the re-scaled values, μ is the mean, and σ is the standard deviation.

Assessing and validating an autoencoder model is very challenging without the presence of labels indicating whether a trajectory is an anomaly or not. For this application, no such labels exist for the extracted OpenSky Network trajectory data, nor the ASOS or ASPM data. Therefore, it is often by manual inspection of the identified anomalies and nominal data that judgements are made regarding the performance and validity of a model. Manually assessing each trajectory is not feasible considering the number of trajectories within the data set. Therefore, to perform some validation, it is ideal to be able to assess quantitative metrics that provide insight into the performance of the model. However, deriving quantitative metrics is difficult without a-priori knowledge of the labels of trajectories.

To address this issue, generation of *artificial anomalies* is proposed as a solution to insert trajectories with labels into the data set. The objective in generating artificial anomalies is to provide the autoencoder with trajectories that were known to be anomalous to assess its capability in detecting these trajectories as anomalous. Two primary types of artificial anomalies were generated:

1) **Extreme Artificial Anomalies (ExAA)**: These were trajectories with nominal latitude and longitude coordinates, yet all other metrics had the potential to be varied to extreme values. Extreme values in this case are either the maximum/minimum values (0% higher/lower than the maximum/minimum values), 5% higher/lower than the maximum/minimum values, and 10% higher/lower than the maximum/minimum values of each metric. A sample ExAA trajectory with the nominal latitude and longitude values was then populated with extreme values for all other metrics, where different combinations of metrics being high extreme or low extreme were created. For each of the extreme value settings (0%, 5%, and 10%), 130 ExAA were generated, for a total of 390 ExAA. The ExAA simulated entire trajectories that were anomalous.

2) **Subsequence Artificial Anomalies (SubAA)**: These were trajectories selected at random, where a subsequence of these trajectories' time series metrics was then also selected at random, and the values of all time series metrics were perturbed to be significantly greater with random noise than the current values. The perturbed values were generally always much higher than the normal range for that time series metric. The metadata metrics were left unvaried in this case. There were 300 SubAA generated in total. The SubAA simulated only portions of entire trajectories that were anomalous, and may be considered more instantaneous anomalies.

A total of 690 artificial anomalies were generated, making up about 0.8% of the total data set. Plots of the energy profiles of the two types of artificial anomalies are displayed in Figure 7 and Figure 8. The lighter green area in the

Figure 7 and Figure 8 represents the range of the 5th to 95th quantiles, while the darker shade of green represents the 25th to 75th percentiles. It is evident these artificially anomalous trajectories are significantly different from the trajectories that exist in the real data set. Thus, the autoencoder should be able to detect dissimilarities between these and real, nominal data. It is assumed that a robust anomaly detection model will make a significant distinction between the artificial anomalies and real data.

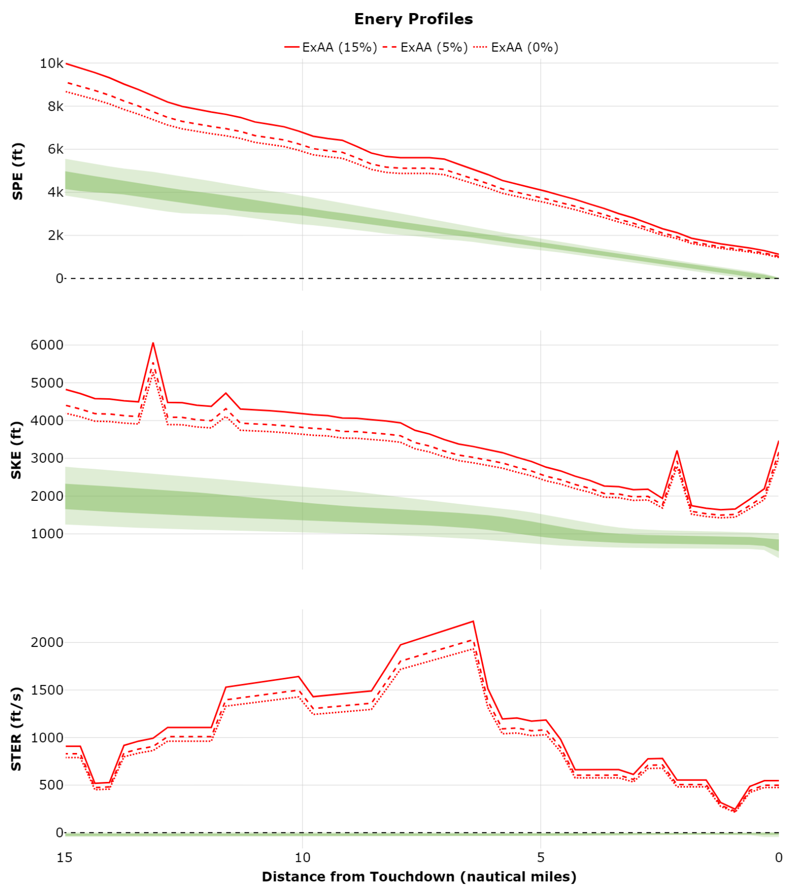

Fig. 7 ExAA examples

D. Post-Processing and Visualization

Anomalies are detected leveraging the anomaly score, or reconstruction error, from the autoencoder. It should follow that the autoencoder is able to reconstruct nominal trajectories well, while abnormal, or anomalous trajectories are poorly reconstructed. A poor reconstruction, thus, results in a higher reconstruction error, or anomaly score. Therefore, the histogram of anomaly scores is considered such that a specified threshold may be set above which the flight segments may be considered to be anomalous. The 99^{th} percentile of the reconstruction errors was selected as the threshold above which trajectories were considered to be anomalous.

Implementation of the previously described autoencoder resulted in 72% of the artificial anomalies being above the 99^{th} percentile value, and therefore classified as anomalies. Further, the distribution of the log of the reconstruction errors (anomaly scores) separated by real data and type of artificial anomaly is displayed in Figure 9. It is evident that the distributions of anomaly scores between both types of artificial anomalies and the real data are clearly dissimilar. The ExAA anomalies appear to be generally "more severe" than the SubAA anomalies, which is consistent with their development, as the ExAA anomalies are dissimilar for the entire trajectory length as well as metadata metrics.

Additionally, it is investigated if there is a clear distinction between the artificial anomalies and real data in the latent space layer. Utilizing the t-distributed stochastic neighbor embedding (t-SNE) [34] dimensionality reduction technique

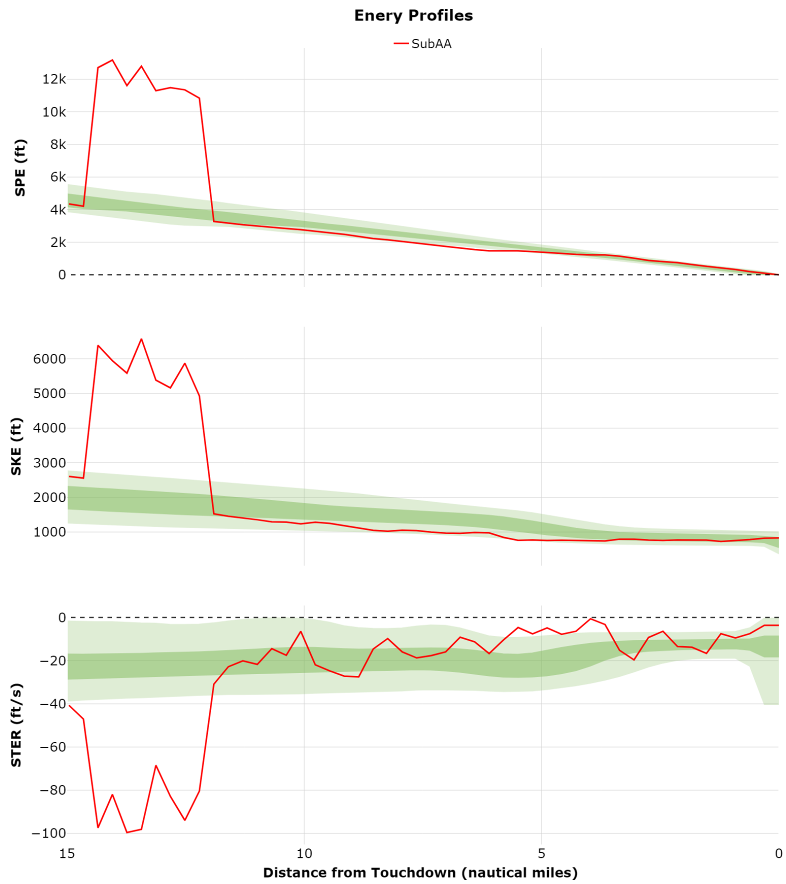

Fig. 8 SubAA example

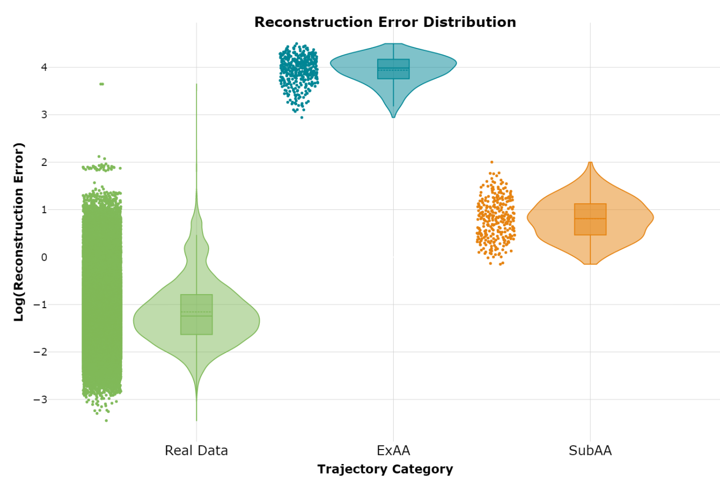

Fig. 9 Model reconstruction error distributions separated by label

to reduce the dimension of the latent space and visualize the latent space, it is clear there is a distinction. The t-SNE dimensionality reduction technique is a non-linear dimensionality reduction techinique, which models the probability distribution of *neighbors* around each points, where the term neighbors refers to the set of points closest to each point. In the original, high-dimensional space this is modeled as a Gaussian distribution, while in the low-dimensional output space this is modeled as a t-distribution. The reduced-dimension plot is displayed in Figure 10. Both the ExAA and SubAA lie on the outskirts of the reduced-dimensional space.

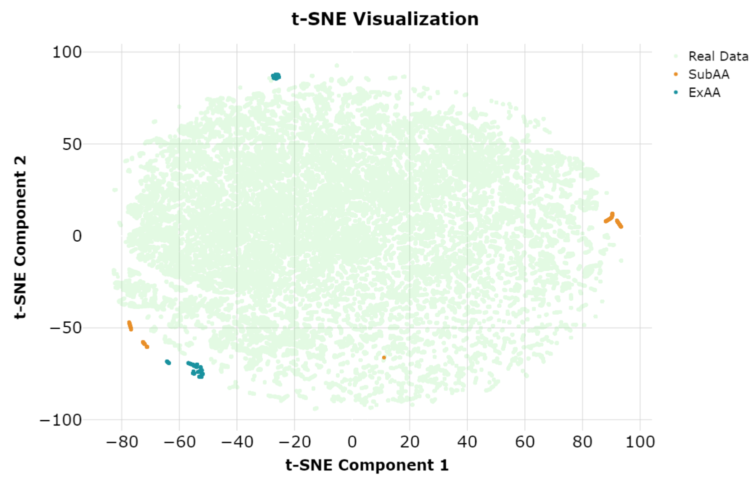

Fig. 10 Model reduced-dimensional plot

Fig. 11 Top anomalies identified

To perform some manual validation on this model, two of the real trajectories with the highest anomaly scores are examined. Figure 11 presents the energy profiles of these two trajectories. It is evident there is abnormal behavior as these trajectories, at times, are beyond the normal ranges of energy metrics. For instance, Anomaly 1 has a very low SKE profile. Anomaly 2 also has a relatively low SKE profile, as well as SPE profile.

The above visualizations developed and discussed enabled an initial validation of the proposed framework to be performed. It is evident that abnormal flights are able to be detected, specifically, as demonstrated in Figure 11, in the energy dimensions. Considering the approach phase, proper energy management is paramount to the safe and efficient arrivals of aircraft.

V. Conclusions and Future Work

This paper presented a novel framework based on deep learning methods, specifically autoencoders, to detect anomalies in terminal airspace operations. In development and implementation of this framework on six months of data for aircraft arriving at KSFO, several key accomplishments related to anomaly detection using fused trajectory data were observed. An extensive data processing pipeline that involved the fusion of data from multiple sources to include information about not only the trajectory of each aircraft, but also the prevailing weather and traffic congestion metrics at an airport was developed and implemented. Artificial anomalies were generated to aid in model validation efforts. Finally, the implementation of an architected deep autoencoder model demonstrated success based on initial validation studies.

It is noted that no hyperparameter optimization was performed in this initial demonstration implementation of the proposed framework. Therefore, future work includes the exploration of hyperparameter optimization to enable selection of the most robust anomaly detection model. Further, the application of this framework to a data set of departing flights is necessary. Based on the initial validation of results presented in Section IV, it is evident the proposed framework shows promise in detection of anomalies in historical trajectory data fused with weather and traffic metrics. In the future, the developed framework has the potential to aid air traffic controllers in identifying high risk situations from a holistic perspective and applying appropriate mitigation strategies.

Acknowledgements

This work was funded by the National Aeronautics and Space Administration (NASA) through the Small Business Innovation Research (SBIR) grant to Global Technology Connection (Award Number 80NSSC20C0481). The authors would like to thank NASA technical monitor Mr. Ryan Chartrand for his valuable feedback and comments. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding entities.

References

[1] "Federal Aviaition Administration Aerospace Forecasts Fiscal Years 2016-2036." https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/FY2016-36_FAA_Aerospace_Forecast.pdf, 2017. Retrieved: 10/2019.

[2] Federal Aviation Administration, "Modernization of the U.S. Airspace," https://www.faa.gov/nextgen/, Sep. 2019.

[3] Single European Sky ATM Research, "About SESAR," https://www.sesarju.ed/discover-sesar/history, 2020.

[4] Das, S., Matthews, B. L., Srivastava, A. N., and Oza, N. C., "Multiple Kernel Learning for Heterogeneous Anomaly Detection: Algorithm and Aviation Safety Case Study," *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2010, pp. 47–56. doi:10.1145/1835804.1835813.

[5] Li, L., Das, S., John Hansman, R., Palacios, R., and Srivastava, A. N., "Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations," *Journal of Aerospace Information Systems*, Vol. 12, No. 9, 2015, pp. 587–598. doi:10.2514/1.I010329.

[6] Basora, L., Olive, X., and Dubot, T., "Recent Advances in Anomaly Detection Methods Applied to Aviation," *Aerospace*, Vol. 6(11), 2019, p. 117. doi:10.3390/aerospace6110117.

[7] Puranik, T., and Mavris, D., "Anomaly Detection in General-Aviation Operations Using Energy Metrics and Flight-Data Records," *Journal of Aerospace Information Systems*, Vol. 15, No. 1, 2018, pp. 22–35. doi:10.2514/1.I010582.

[8] Puranik, T. G., and Mavris, D. N., "Identification of Instantaneous Anomalies in General Aviation Operations using Energy Metrics," *Journal of Aerospace Information Systems*, Vol. Article in Advance, 2019. doi:10.2514/1.I010772.

[9] Logan, T. J., "Error Prevention as Developed in Airlines," *International Journal of Radiation Oncology, Biology, Physics*, Vol. 71, No. 1, 2008, pp. S178–S181.

[10] Puranik, T. G., "A Methodology for Quantitative Data-driven Safety Assessment for General Aviation," Ph.D. thesis, Georgia Institute of Technology, 2018.

[11] Basora, L., and Xavier Olive, T. D., "Recent Advances in Anomaly Detection Methods Applied to Aviation," *Aerospace*, Vol. 6, No. 117, 2019.

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

[13] Li, L., Hansman, J., Palacios, R., and Welsch, R., "Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring," *Reliability Engineering & System Safety*, Vol. 193, 2020.

[14] Sridhar, B., "Applications of Machine Learning Techniques to Aviation Operations: Promises and Challenges," *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, 2020, pp. 1–12. doi:10.1109/AIDA-AT48540.2020.9049205.

[15] Olive, X., and Morio, J., "Trajectory clustering of air traffic flows around airports," *Aerospace Science and Technology*, Vol. 84, 2019, pp. 776–781.

[16] Puranik, T., Jimenez, H., and Mavris, D., "Energy-Based Metrics for Safety Analysis of General Aviation Operations," *Journal of Aircraft*, Vol. 54, No. 6, 2017, pp. 2285–2297.

[17] Kim, K., and Hwang, I., "Terminal Airspace Anomaly Detection Using Temporal Logic Learning," 2018.

[18] Deshmukh, R., "Data-Driven Anomaly and Precursor Detection in Metroplex Airspace Operations," Ph.D. thesis, Purdue University, 2020.

[19] Corrado, S., Puranik, T., Pinon, O., and Mavris, D., "Trajectory Clustering within the Terminal Airspace Utilizing a Weighted Distance Function," *Proceedings of the 8th OpenSky Symposium 2020*, 2020, pp. 1–10. doi:https://doi.org/10.3390/proceedings2020059007.

[20] Jarry, G., Delahaye, D., Nicol, F., and Feron, E., "Aircraft atypical approach detection using functional principal component analysis," *Journal of Air Transport Management*, Vol. 84, 2020.

[21] Olive, X., and Basora, L., "Detection and Identification of Significant Events in Historical Aircraft Trajectory Data," *Transportation Research Part C: Emerging Technologies*, Vol. 119, 2020.

[22] Corrado, S. J., Puranik, T. G., Pinon-Fischer, O., and Mavris, D. N., "A unified, clustering-based framework for detection of spatial and energy anomalies in trajectories utilizing ADS-B trajectory data," , 2021. `https://www.preprints.org/manuscript/202102.0396/v1`.

[23] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly detection: A survey," *ACM Computing Surveys*, Vol. 41, No. 3, 2009.

[24] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature*, Vol. 521, No. 7553, 2015, pp. 436–444.

[25] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.

[26] An, J., and Cho, S., "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, Vol. 2, No. 1, 2015.

[27] Olive, X., and Bieber, P., "Quantitative Assessments of Runway Excursion Precursors using Mode S data," *ICRAT - International Conference for Research in Air Transportation*, 2018.

[28] Olive, X., and Basora, L., "Identifying Anomalies in past en-route Trajectories with Clustering and Anomaly Detection Methods," *Thirteenth USA/Europe Air Traffic Management Research and Development Seminar*, 2019.

[29] Olive, X., and Basora, L., "Detection and Identification of Significant Events in Historical Aircraft Trajectory Data," *Transportation Research Part C: Emerging Technologies*, Vol. 119, 2020.

[30] Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M., "Bringing Up OpenSky: A Large-scale ADS-B Sensor Network for Research," *13th IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPNS)*, 2014, pp. 38–94.

[31] Olive, X., "Traffic, a toolbox for processing and analysing air traffic data," *Journal of Open Source Software*, , No. 7553, 2019, p. 1518.

[32] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," , 2015. URL https://www.tensorflow.org/, software available from tensorflow.org.

[33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[34] van der Maaten, L., and Hinton, G., "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, Vol. 9, No. 86, 2008, pp. 2579–2605. URL http://jmlr.org/papers/v9/vandermaaten08a.html.