



Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc



Temporal logic learning-based anomaly detection in metroplex terminal airspace operations

Raj Deshmukh*, Dawei Sun, Kwangyeon Kim, Inseok Hwang

School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907, United States



ARTICLE INFO

Keywords:

Air traffic management
Metroplex operations
Air traffic surveillance data
Airport operations
Machine learning
Anomaly detection

ABSTRACT

The airspace system is a complex dynamical system with complicated controlled interactions between its constituent subsystems – terminal airspace, en-route airspace, and ground. Of these, air traffic management in the multi-airport (metroplex) terminal airspace is one of the most complicated subsystems to manage, especially due to the interactions between proximal airports. Analyzing anomalous behaviors in the metroplex is emerging as a key problem in understanding air traffic management complexity and safety. Although physics-based approaches have been studied in-depth for this application, newfound interest has been observed to use recorded time-series air traffic surveillance and airport operations datasets for this purpose. In this paper, we propose a machine learning-based anomaly detection algorithm that generates mathematical models to detect anomalies in metroplex operations. Several machine learning algorithms have been developed to detect anomalies using only air traffic surveillance data, but there is a significant scope of improvement by including airport operational characteristics as well, since integrating such closely-controlled metroplex operational datasets allows the developed models to effectively detect anomalies. The key contribution of this paper is in allowing anomaly detection models to recursively update so as to adapt to changes in metroplex operations. The proposed algorithm is demonstrated with real air traffic surveillance and airport operations datasets at LaGuardia, John F. Kennedy, and Newark airports, thereby detecting anomalies for operations in the New York metroplex.

1. Introduction

The air traffic management (ATM) system is one of the most complex and continuously-evolving systems, where a number of air traffic controllers (ATCs) coordinate and control navigation of flights. The global air traffic density doubles roughly every 15 years (IATA, 2019), and the growth is expected to speed up with increasing public acceptance, liberalization, and improved ATM technologies. Modernization programs of the FAA and NASA, such as NextGen (Next Generation Air Transportation System) (Modernization of U.S. Airspace, 2019) and SMART-NAS (Shadow Mode Assessment Using Realistic Technologies for the National Airspace System) (SMART-NAS, 2017) have focused on meeting future operational capacity and efficiency, while ensuring and enhancing safety in the highly coordinated National Airspace System (NAS). Among the various components of the NAS, such as en-route, terminal, and surface operations, the higher traffic density and highly structured operational policies make the operations in terminal airspace have a

* Corresponding author.

E-mail addresses: rdesmu@purdue.edu (R. Deshmukh), sun289@purdue.edu (D. Sun), kim1601@purdue.edu (K. Kim), ihwang@purdue.edu (I. Hwang).

higher impact on the entire system's safety (about 75% of incidents in the NAS) (Oster et al., 1992; Xue and Zelinski, 2014). Of these, the terminal airspace of a metropolitan area where multiple closely-located airports coordinate with each other, called *metroplex* (Clarke et al., 2012), such as in New York, Houston, Denver, and Las Vegas, is key in ensuring and enhancing the safety in the entire NAS, due to its even higher traffic density and more strictly structured coordination policies between the multiple airports. In such a complex and safety-critical system, it is important to identify degradations in system performance. Especially in terms of safety, such degradations in airspace operations could correspond to issues involving multiple aircraft (e.g., conflicts between aircraft, or loss of separation) or issues for a single aircraft (e.g., proximity to the ground, passenger discomfort, fatigue for the airframe, etc.) (Li et al., 2011; A.C.A.S. Authority, 2011; Gallego et al., 2019). The objective of this paper is to develop an algorithm to detect such degradations, called *anomalies*, in metroplex operations, which allows us to get an insight into understanding its complex behaviors, and that can be used by domain experts and policymakers to prevent or mitigate such anomalous behaviors in the future.

There are two main approaches in the topic of anomaly detection in aviation: (i) *Physics-based approaches*: In these approaches, a model is developed to characterize the normal behavior of a dynamical system based on governing physics, followed by an algorithm to detect specific occurrences that do not conform to the model, such as the thermodynamic model of the engine (Narasimhan and Brownston, 2007) and the dynamics of an aircraft (Zhu et al., 2016); and (ii) *Data-driven approaches*: With the recent advent of sensing technologies and data collection capabilities that allow abundant recordings of aviation data, modern machine learning techniques can provide powerful tools for anomaly detection in aviation applications. These approaches seek to generate a data-driven anomaly detection model that separates anomalies from normal data (Agrawal and Agrawal, 2015; Gavrilovski et al., 2016). In some applications of machine learning, the data already has labels tagging the anomalous and normal data, and the algorithms generate a model to separate the labeled data. This class of machine learning algorithms, called *supervised learning* algorithms, uses techniques such as supervised neural networks (Wan, 1990), support vector machines (SVMs) (Cortes and Vapnik, 1995), k-nearest neighbors (Zhang and Zhou, 2005), Bayesian networks (Friedman et al., 1997), and decision trees (Ghorbani et al., 2009). In aviation applications, regression-based supervised learning algorithms have been employed to model aerodynamic force data (Gorinevsky et al., 2012) and to detect anomalies in engine fuel consumption (Srivastava, 2012). On the other hand, aviation data is typically unlabeled, i.e., there is often no information implying whether a specific data is anomalous or normal, and thus, supervised learning algorithms cannot be directly used. In this case, *unsupervised learning* algorithms are used, which find large clusters of data with similar characteristics, and thus the outliers that are not included in the clusters are identified as anomalies (Hodge and Austin, 2004). These outliers are deviations from normal behaviors, and thus typically have an operational or a safety-related impact. Not all identified anomalies may be actual threats: some of them may significantly violate expected normal behavior, while others only slightly deviate from expectation. Thus, unsupervised learning algorithms rely on qualitative feedback from subject matters experts for improving performance. Some unsupervised anomaly detection techniques like clustering rely on a distance or density based approach, where the algorithm infers anomalies based on the distribution of the data itself, by finding those data points which are at a greater distance from most of the other data points or by finding those data points which are in a low density region (Ester et al., 1996; Das et al., 2011; Dani et al., 2015; Shin and Hwang, 2017). In Li et al. (2016), the authors use a Gaussian Mixture Model (GMM) based clustering of digital flight data to detect anomalous flights and simultaneously identify the abnormal part of the detected flight. A multiple kernel learning approach for heterogeneous data, called Multiple Kernel Anomaly Detection (MKAD) algorithm, is proposed in (Das et al., 2010) to detect anomalies in the Flight Operational Quality Assurance (FOQA) archives. The MKAD approach has been applied to other classes of aviation anomaly detection problems, such as detecting anomalies in human-machine interactions (Vaidya et al., 2016) and in routine airline operations (Li et al., 2015). The one-class support vector machine (OCSVM) based methods attempt to design a hyperplane that has all normal data on one side (Ma and Perkins, 2003) and have been applied to anomaly detection in propulsion health monitoring (Schwabacher et al., 2009) and in energy metrics for general aviation operations (Puranik and Mavris, 2017).

Most of the aforementioned data-driven algorithms do not consider the implications of operational conditions in the airspace, i.e., they only find anomalies in single-aircraft behaviors. However, embedding domain knowledge of the operational conditions in the metroplex terminal airspace is crucial to enhance the performance and effectiveness of anomaly detection. Moreover, airspace structure evolves over time, even within a single day, and thus, a recursively or incrementally learned anomaly detection model that can efficiently ingest and assimilate newly recorded (incremental) data would be more appropriate. Thus, satisfying these requirements in an anomaly detection algorithm will make it more applicable to anomaly detection in the metroplex operations.

In this paper, we propose an incremental learning algorithm that detects anomalies in the metroplex operations, using air traffic surveillance data representing the individual aircraft behaviors along with data representing airport operational conditions. Our algorithm is recursive, in the sense that it ingests newly recorded data and incrementally adapts its anomaly detection models so that the models can effectively detect anomalies not only at a single airport but also within an entire closely coordinated metroplex airspace. To the best of our knowledge, there has been no prior research dealing with metroplex airspace for anomaly detection, which we present in this paper by incorporating the intricate coordination and operational policies between proximal airports.

The rest of the paper is organized as follows: in Section 2, the proposed anomaly detection algorithm (which we call *event-triggered TempAD*) is described. This is followed by the results of extensive tests and analysis of the proposed algorithm with real aviation data in Section 3. Finally, concluding remarks are made in Section 4.

2. Algorithm development

This paper aims to develop an algorithm for generating anomaly detection models for the metroplex operations, where *anomaly detection models* are defined as the mathematical representation of normal behaviors. The normal behaviors in the metroplex operations can be characterized in two aspects: (i) individual aircraft's behavior following the procedures at a single airport (Fig. 1(a)); and

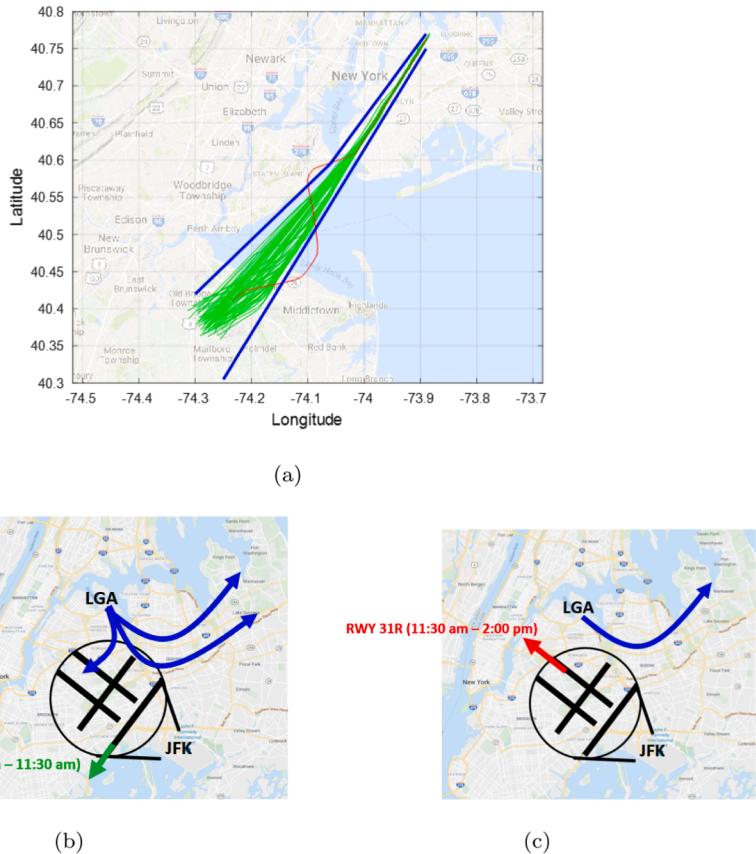


Fig. 1. Normal behaviors in metroplex operations: (a) Trajectories of normal arrival flights to LGA (in green); (b) Collective normal climb paths (in blue) for flights departing from LGA when JFK is operating RWY 22L; and, (c) Collective normal climb path (in blue) for flights departing from LGA when JFK is operating RWY 31R.

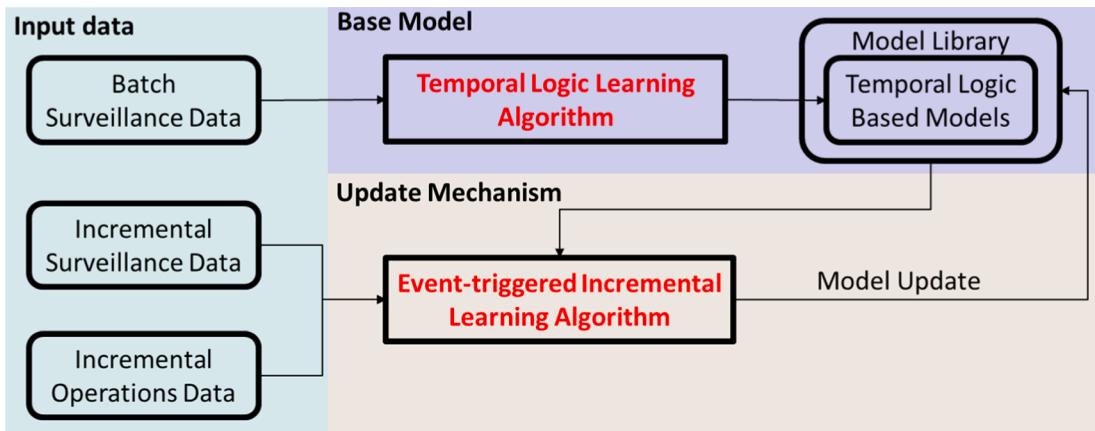


Fig. 2. Overarching framework of the proposed anomaly detection algorithm.

(ii) collective aircraft behaviors according to the multi-airport coordination policies (Fig. 1(b) and (c)). In this regard, the proposed algorithm ingests a heterogeneous input: (i) a surveillance dataset which records individual aircraft's states such as position and speed in a time-series format; and (ii) an operations dataset which contains information such as the operating runway, arrival and departure exit gates, and multi-airport coordination policies, which are discussed in Section 2.1.

Since all these datasets keep continuously recording daily operations, it is desirable for the generated anomaly detection models to appropriately incorporate this newly recorded incremental data, but also to keep the knowledge learned from the past dataset. One

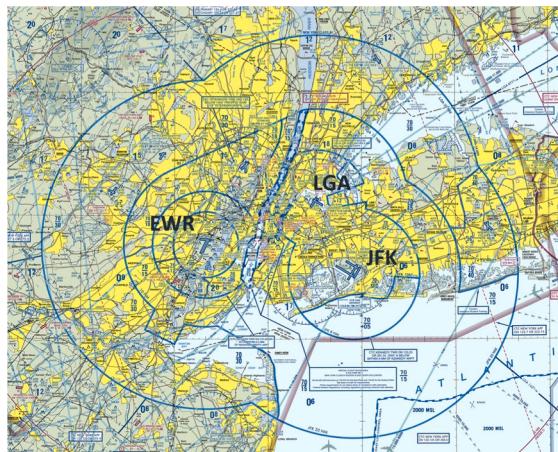


Fig. 3. Shared terminal airspace of major airports in the New York metroplex: LGA, JFK and EWR.

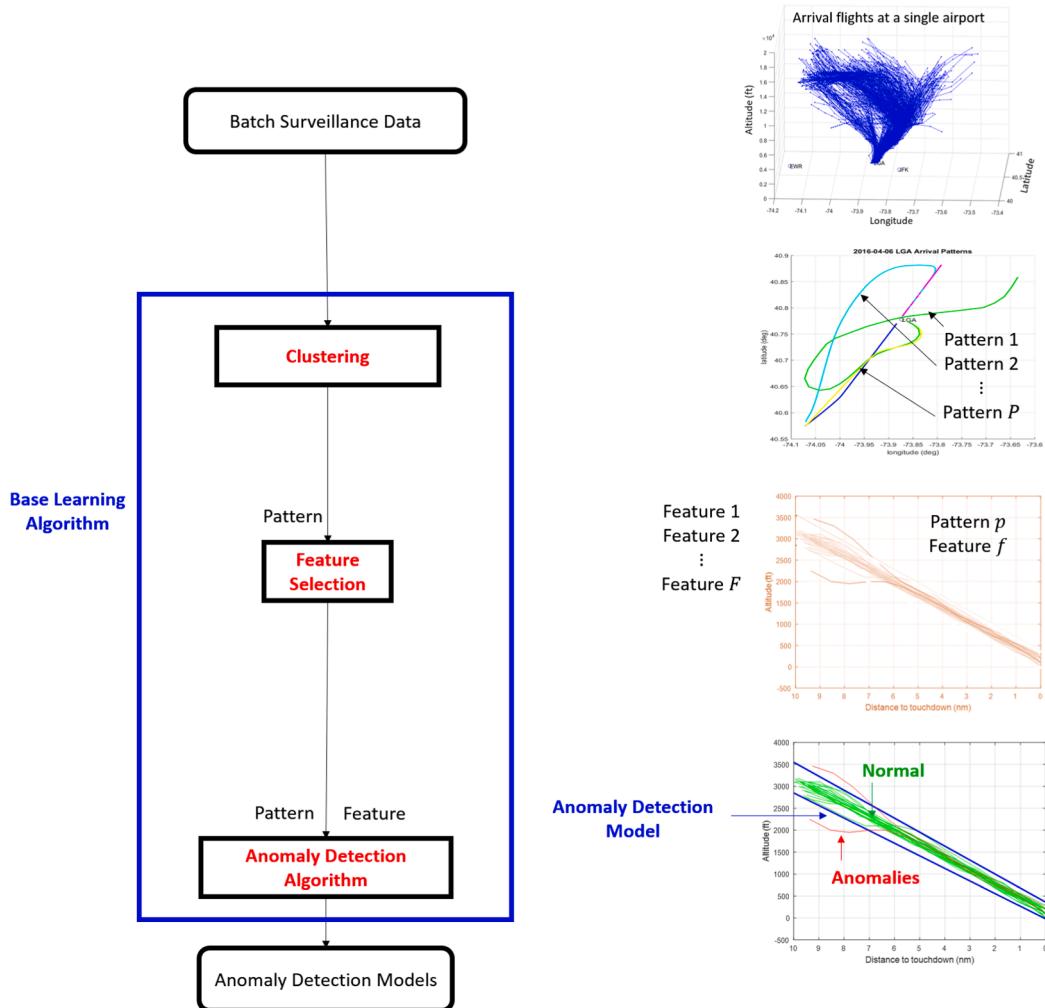


Fig. 4. Base learning algorithm.

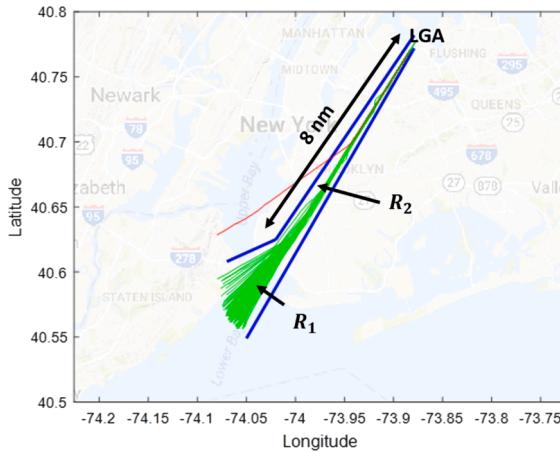


Fig. 5. Horizontal anomaly detection model for arrivals to RWY4 at LGA.

way to achieve this is to update the past dataset with the incremental data and then learn an entirely new model from the updated dataset (called *batch surveillance data*). With this approach, any machine learning algorithm (which we call the *base learning* algorithm and discuss in Section 2.2) can be directly applied to the updated dataset without any modifications. However, the computational complexity of the base learning algorithm grows drastically and the generated anomaly detection models tend to be ineffective as the size of the dataset continually grows. Another way to incorporate the incremental data is to maintain and update a *model library* (a set of learned anomaly detection models) instead of updating the entire dataset, considering the models before the update as a representative summary of the past dataset. This approach, called *incremental learning*, requires solutions for two sub-problems: (i) when to update the model and (ii) how to update the model. Between the two ingested datasets, the operations dataset plays an important role in the first sub-problem (when to update). The operations data represents the collective behaviors of the aircraft in the metroplex. Therefore, rather than updating the model library in a temporal fashion, e.g., every day (Deshmukh and Hwang, 2019), it would be more effective to *trigger* the update only when we detect any change in the aircraft's collective behavior, called *event* in this paper, e.g., change in the operating runway. The surveillance dataset is key in solving the second sub-problem (how to update). Based on how similar (or dissimilar) the models before the update are to the incremental surveillance data, the amount of required update (small, large or extremely large) is determined and the model library is then updated accordingly, which is discussed in Section 2.3. Thus, we propose an *event-based incremental learning* algorithm, as shown in Fig. 2. In the following subsections, the details of the (i) heterogeneous input, (ii) base learning algorithm, and (iii) event-based incremental learning algorithm are presented.

2.1. Input data

As discussed above, the proposed algorithm takes as input two datasets: the surveillance dataset and operations dataset. In this paper, we use the Terminal Automation Information Service (TAIS) (STDDS, 2017) dataset and the Aviation System Performance Metrics (ASPM) (ASPM, 2019) dataset as the surveillance and operations datasets, respectively. The TAIS dataset has a detection range of about 141 nautical miles from an airport and records aircraft states, such as latitude, longitude, altitude, and speed, every 5 s. The ASPM data records parameters at airports such as operating runway, delay counts, windspeed, and ceiling, every quarter hour. All these datasets used in this paper were recorded from September to November in 2016, in the New York metroplex (shown in Fig. 3), whose major airports are the LaGuardia airport (LGA), John F. Kennedy airport (JFK), and Newark airport (EWR).

2.2. Base learning algorithm

For each airport in the metroplex, we first establish a model library from the batch surveillance data via a base learning algorithm (Kim and Hwang, 2018; Deshmukh and Hwang, 2019). As shown in Fig. 4, the base learning algorithm consists of three parts: (i) clustering, (ii) feature selection, and (iii) anomaly detection algorithm.

Clustering. Even for a single airport, operations are very complex and involve multiple climb paths for departures and approaches for arrivals, as shown in the leftmost in Fig. 4. Generating anomaly detection models on the entire dataset at once can yield too conservative models to detect some important anomalous flights whose states are buried within the entire dataset. To address this conservativeness, if we examine only the flights with similar properties, the anomaly detection models can be generated more accurately and efficiently. It is observed that the horizontal trajectories can characterize the flights, i.e., every flight taking the same (or similar) horizontal trajectory has similar properties such as speed and flight path angle. Hence, the batch surveillance data is first pre-processed by performing clustering using the popular Density-based Spatial Clustering of Applications with Noise (DBSCAN) technique (Ester et al., 1996) along the horizontal dimension (i.e., using latitude and longitude) to identify the patterns, as shown in the second-from-left plot in Fig. 4.

Feature selection. For each identified pattern, we then select specific features corresponding to the properties or states of the flight to

use as input to the anomaly detection algorithm. From the surveillance dataset, we first select the basic features: the horizontal (latitude x and longitude y), vertical (altitude h), and speed (speed v). We then synthesize the derived features from the basic features: the specific total energy (STE, $h + v^2/2g$) and the specific potential energy rate (SPER, \dot{h}). Here, g is the acceleration due to gravity (9.81 m/s^2). These energy features are used because energy management is important for aircraft operations, especially in terminal airspace, so that energy excess or energy deficit anomalies (Puranik and Mavris, 2017) can be better detected.

Anomaly detection algorithm. Suppose there are N flights within a given pattern. We denote the set of time-series of a feature as $\{s_i\}_{i=1}^N$ for the i -th flight. Since the time-series data is unlabeled (i.e., it is unknown which flight is normal or abnormal), this naturally leads to applying an unsupervised learning approach. Once the model is generated in an unsupervised manner, the model can be further improved through feedback from a subject matter expert (SME) about whether the detected anomalies are operationally significant or not. Hence, the models should be easily interpretable by the SME. For example, the model (in blue) in Fig. 5 can be interpreted in natural language as ‘any approaching aircraft should reside in the region 1 (R_1) until 8 nautical miles from touchdown, and then reside in the region 2 (R_2) until touchdown.’ This interpretation can be translated into a mathematical expression:

$$G_{[t_0, t_1]}((x, y) \in R_1) \wedge G_{[t_1, t_2]}((x, y) \in R_2) \quad (1)$$

where x and y are the longitude and latitude, respectively, G is the global operator (implying ‘always’), and \wedge is the logical ‘and’ operator. The subscripts $[t_0, t_1]$ and $[t_1, t_2]$ represent the bounds for a temporal parameter t , which is the remaining distance to touchdown in this example. The regions R_1 and R_2 are characterized by:

$$\begin{aligned} R_1 &= \{(x, y) : 0.411x + y - 71.0476 < 0 \text{ and } 1.3041x + y - 37.1171 > 0\} \forall t \in [10, 8] \\ R_2 &= \{(x, y) : 1.1135x + y - 123.0444 < 0 \text{ and } 1.3041x + y - 37.1171 > 0\} \forall t \in [8, 0] \end{aligned} \quad (2)$$

Expression (2) is referred to as a *temporal logic model* (Kong et al., 2017), which consists of two parts, as shown in the example: (i) structure of the model (such as piecewise linear polynomials and the number of pieces) and (ii) parameters of the model (such as a , b and c in $ax + by + c < 0$). The approach to constructing such a temporal logic model expression in order to distinguish between normal and anomalous behavior is called *temporal logic learning*.

The structure (or shape) of the model can be determined by finding the locations at which the recorded data can be separated into sections (or different linear polynomials), e.g., R_1 and R_2 . Considering that aircraft follow consistent patterns according to ATC regulations, the collective aircraft behaviors can be approximately represented by the centroid (mean) of states of all the flights that follow a given pattern. We determine the model structure using piecewise regression to find the best piecewise linear fit to this centroid.

For the time-series dataset $\{s_i\}_{i=1}^N$, the parameters of the anomaly detection model, denoted by φ_θ (where θ refers to the model parameters), can be determined by optimizing the following cost function:

$$\min_{\theta, \epsilon} f(\{s_i\}_{i=1}^N, \varphi_\theta) \quad (3)$$

where $f(\{s_i\}_{i=1}^N, \varphi_\theta) = d(\{s_i\}_{i=1}^N, \varphi_\theta) + (-\epsilon) + \frac{1}{\alpha N} \sum_{i=1}^N \mu(s_i, \varphi_\theta)$

Here,

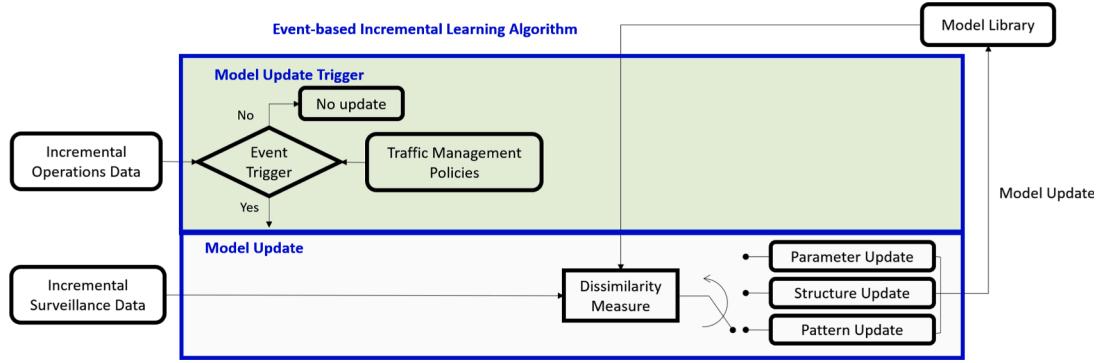
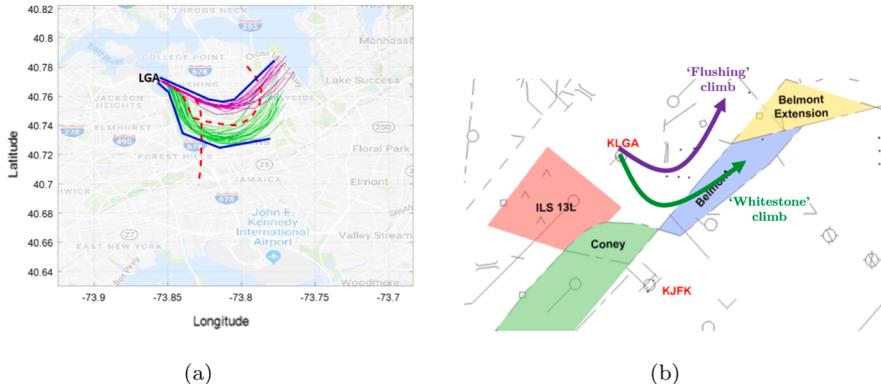
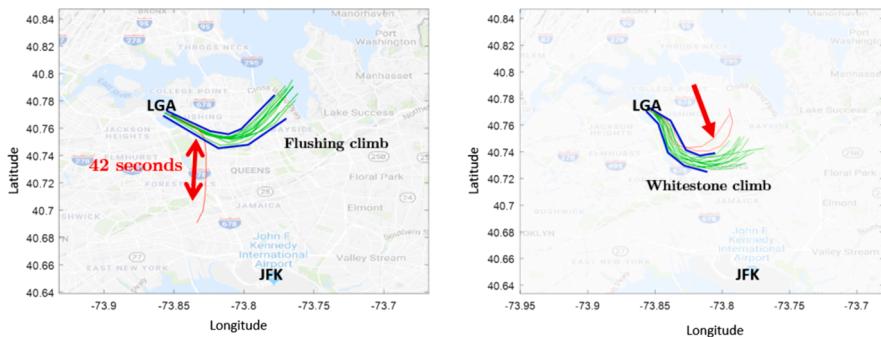
- $d(\{s_i\}_{i=1}^N, \varphi_\theta)$ is a tightness function (Jones et al., 2014) that penalizes the conservativeness of the model. Minimizing d prevents the learned model φ_θ from trivially describing all observed time-series data. During the implementation of the proposed algorithm, we chose the tightness functional value for a candidate anomaly detection model as the cumulative absolute distance between the model and the centroid of all trajectories at each nautical mile.
- ϵ is the gap between normal and anomalous time-series data. Maximizing ϵ makes the model robust to noise in the input data.
- $\mu(s_i, \varphi_\theta)$ is a slack variable, designed as a hinge-loss function which is positive if s_i does not satisfy φ_θ with minimum robustness of ($\epsilon/2$).

$$\mu(s_i, \varphi_\theta) = \begin{cases} 0, & \text{if } r(s_i, \varphi_\theta) > \frac{\epsilon}{2} \\ \frac{\epsilon}{2} - r(s_i, \varphi_\theta), & \text{otherwise} \end{cases}$$

where $r(s_i, \varphi_\theta)$ is called the robustness degree (Fainekos and Pappas, 2009; Kong et al., 2014), and is the signed distance of the time-series data s_i from the model φ_θ (designed to give negative values for anomalous data and positive values for normal data). This hinge-loss term minimizes the number of time-series data that the model φ_θ classifies as anomalous.

- α is a tuning parameter for trade-off: choosing a higher value of α results in a tighter bound and choosing a lower value of α results in a conservative bound.

During the implementation of the proposed algorithm, the robustness degree value for each signal is computed as the maximum perpendicular distance between the learned model and the signal.

**Fig. 6.** Update mechanism for event-triggered TempAD.**Fig. 7.** Impact of multi-airport operational conditions on departures from LGA RWY13: (a) Horizontal anomaly detection model for all departures; and, (b) Structure of shared airspace between LGA (ICAO code: KLGA) and JFK (ICAO code: KJFK).**Fig. 8.** Horizontal anomaly detection models for all departures from LGA RWY13, segregated by climb paths (Left: 'Flushing' climb, right: 'Whitestone' climb).

The robustness degree, $r(s_i, \varphi_0)$, has a more negative value if a flight violates the model more severely. By normalizing this r between 0 (least anomalous) and 1 (most anomalous), a degree of anomalousness, denoted by \hat{r} , can be computed for each anomalous flight. This \hat{r} describes the significance of the detected anomaly, and can help make a good choice for α and judge the quality of the learned anomaly detection models.

The temporal logic learning (for structure and parameters) for model generation is applied to all the patterns and features, and the generated anomaly detection models comprise the initial model library.

2.3. Event-based incremental learning algorithm

Next, we need to update the model library in order to incorporate the incremental data as well as keep the knowledge learned from

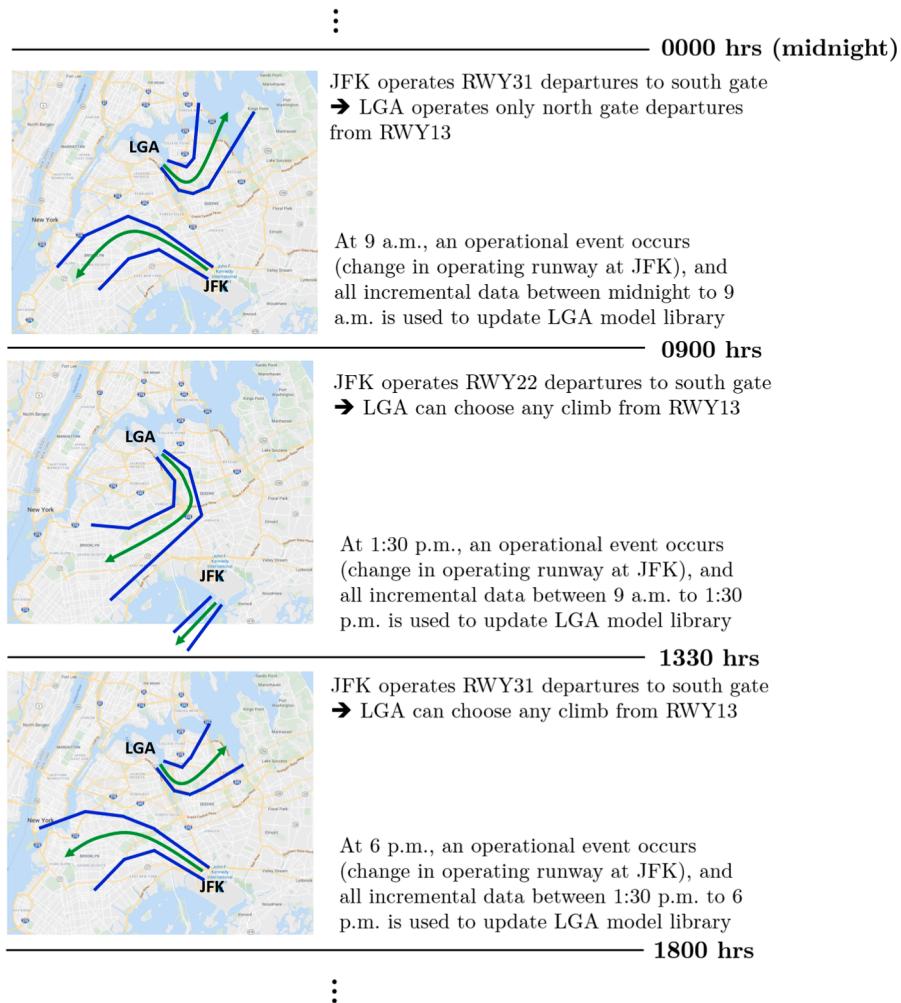


Fig. 9. Event triggers for updating LGA model library.

the past dataset. For this purpose, we propose an *event-based incremental learning algorithm*, which consists of two parts: (i) *model update trigger* for ‘when to update’; and (ii) *model update* for ‘how to update’, as shown in Fig. 6, which are presented in details in what follows.

Model update trigger: Operational event triggers. As a motivating example, consider the departures from runway 13 at LGA during a day, as shown in Fig. 7(a). The anomaly detection model (in blue) is generated for all the departures throughout the entire day, and it detects two anomalies (in red). However, it is easily observed that there are two distinct climb paths (in green and purple) and the model does not accurately represent each climb path, i.e., it is too conservative.

Taking into account not only the operating runway at LGA, but also its combination with the operating runway at JFK, we can generate more accurate models, as shown in Fig. 8. Initially from 7:00 am to 8:59 am, both ‘Coney’ and ‘Belmont’ airspaces (shown in Fig. 7(b)) are required by JFK for its departure operations, and thus only ‘Flushing’ climb (in purple) is allowed for the departures at LGA. Later, from 9:00 am to 12:59 pm, JFK gives up ‘Belmont’ airspace, which allows the departures at LGA to use ‘Whitestone’ climb (in green). Using this knowledge about the metroplex operations, we can first generate the model at LGA for ‘Flushing’ climb and when a change in JFK operating runways is observed, we can then update the model library for ‘Whitestone’ climb, thus generating two different models that can more accurately describe the metroplex operations, leading to more effective anomaly detection. These updated models enable us to detect the anomalies sooner (42 s earlier in this example) for ‘Flushing’ climb departures and detect a new anomaly for ‘Whitestone’ climb departures.

As illustrated by this motivating example, the operations dataset plays an important role in determining when to update. As presented in Section 2.1, we use the ASPM data recorded at the three airports within the New York metroplex to obtain the information about the operations at each airport. From the ASPM dataset, we use the following parameters as features:

- Time of operation: starting from midnight, this information is recorded every quarter hour
- Operating runway: at every quarter hour, which arrival and departure runway configuration is used at the airport

To incorporate the knowledge of multi-airport coordination within the metroplex, we also consider the FAA's strategic policies, or Standard Operating Procedures (SOPs) (FAA, October 2017), as a static input that provide the permissible combinations of operational parameters. A few are listed below:

- JFK: The combination of time of day, the operating runways, and the active arrival and departure gates allowed for JFK-controlled flights. For example, between 10:00 p.m. to 7:00 a.m., if JFK operates runway 22R for departures, all aircraft are restricted to 'Gateway' climb (New York ARTCC - JFK SOP, 2019).
- LGA and EWR: The combination of operating runways at JFK, EWR, and LGA, the active arrival and departure gates, and the arrival or climb path permissible for EWR-controlled and LGA-controlled flights. For example, if JFK uses 'Coney' airspace, then all LGA departures from runway 13 are restricted to 'Whitestone' climb (New York ARTCC - LGA SOP, 2019).

Each change in the operational parameters in the metroplex which can potentially affect the collective aircraft behaviors is an *operational event trigger*, after which the model library is required to be updated. A representative illustration of how the operational events between LGA and JFK relate to the event triggers for horizontal anomaly detection models for departures at LGA is presented in Fig. 9.

Model update: Dissimilarity measure and update types. Once the update is triggered by the operational event trigger, the degree of update required (called *update type*: small, large, and extremely large) is determined based on how well the past models can account for the incremental surveillance data. Recall that for a given model φ_θ , the cost function $f(\{s_i\}_{i=1}^N, \varphi_\theta)$ in Eq. (3) represents (i) how many s_i 's violate the model φ_θ by the tightness function $d(\{s_i\}_{i=1}^N, \varphi_\theta)$, and (ii) how severe each violation is by the slack variable $\mu(s_i, \varphi_\theta)$. Suppose we have a model φ_θ learned from a past dataset $\{s_i\}_{i=1}^N$ and we have a newly recorded incremental dataset $\{s_i\}_{i=N+1}^{N+N'}$ (N' is the number of flights in the incremental dataset). Let $P := f(\{s_i\}_{i=1}^N, \varphi_\theta)$ and $Q := f(\{s_i\}_{i=N+1}^{N+N'}, \varphi_\theta)$ for the past and incremental datasets, respectively. Then, the relative change in f from the past to the incremental datasets, $\left| \frac{P-Q}{P} \right|$, represents how dissimilar the model learned from the past dataset is to the incremental dataset (the larger, the more dissimilar), and is thus called *dissimilarity measure*. Also recall that a model φ_θ is generated for an identified (horizontal) pattern and consists of the structure and corresponding parameters (as presented in Section 2.2). In this regard, we determine the update types according to the value of the dissimilarity measure, as follows:

$$\text{Update type} = \begin{cases} \text{Parameter update} & \text{if } \left| \frac{P-Q}{P} \right| < \text{threshold}_1 \\ \text{Structure update} & \text{if } \text{threshold}_1 < \left| \frac{P-Q}{P} \right| < \text{threshold}_2 \\ \text{New pattern update} & \text{if } \left| \frac{P-Q}{P} \right| > \text{threshold}_2 \end{cases}$$

- If the dissimilarity measure has a small value, we determine that the existing model represents the incremental data well, and thus only an update in parameters (*parameter update*) is triggered.
- If the dissimilarity measure has a large value, we determine that the existing model does not represent the incremental data well enough, and thus the structure of the model needs to be changed (*structure update*).
- If the dissimilarity measure has an extremely large value, we determine that the existing model cannot represent the incremental data at all, and thus a *new pattern update* is triggered and a new model which can accurately account for the incremental data needs to be generated.

The thresholds (threshold_1 and threshold_2) to separate each update type are initialized as 0.25 and 0.75, respectively, and are then dynamically updated in a moving average fashion using the value of Q . Each update type is performed similar to each corresponding part in the base learning algorithm as described in Section 2.2, and further details can be found in our previous work (Deshmukh and Hwang, 2019).

In the next section, the results of anomaly detection using the proposed algorithm, which we call *event-triggered TempAD* (temporal logic based anomaly detection), applied to real air traffic surveillance and airport operations data are presented, followed by an analysis of the results.

3. Test and analysis of event-triggered TempAD algorithm

In this section, we first present a summary of results from extensive tests performed using the proposed event-triggered TempAD algorithm. Then, a comparative study is performed against an algorithm that relies only on the surveillance dataset for anomaly detection. This is followed by illustrative examples that show how the use of the operational dataset for anomaly detection allows the proposed algorithm to detect anomalies more effectively.

Table 1

Distribution of anomalies across departures, detected using event-triggered TempAD.

Feature	Airport	No. of flights	No. of anomalies	Anomalies per 1000 departure flights
Horizontal	LGA	37,112	464	12.5
	JFK	41,391	519	12.5
	EWR	43,778	496	11.3
Vertical	LGA	37,112	148	4
	JFK	41,391	169	4.1
	EWR	43,778	161	3.7
Speed	LGA	37,112	127	3.4
	JFK	41,391	149	3.6
	EWR	43,778	181	4.1
STE	LGA	37,112	136	3.7
	JFK	41,391	110	2.7
	EWR	43,778	158	3.6
SPER	LGA	37,112	195	5.2
	JFK	41,391	246	5.9
	EWR	43,778	217	5

Table 2

Distribution of anomalies across arrivals, detected using event-triggered TempAD.

Feature	Airport	No. of flights	No. of anomalies	Anomalies per 1000 arrival flights
Horizontal	LGA	36,243	638	17.6
	JFK	46,852	696	15.2
	EWR	45,471	661	14.5
Vertical	LGA	36,243	314	8.6
	JFK	46,852	431	9.1
	EWR	45,471	389	8.5
Speed	LGA	36,243	182	5
	JFK	46,852	207	4.4
	EWR	45,471	219	4.8
STE	LGA	36,243	149	4.1
	JFK	46,852	167	3.5
	EWR	45,471	170	3.7
SPER	LGA	36,243	242	6.6
	JFK	46,852	258	5.5
	EWR	45,471	256	5.6

3.1. Summarized test results and illustrative cases

The summarized test results for all the features (the horizontal, vertical, speed, and energy anomalies) for all the departures and arrivals are presented in [Table 1](#) and [Table 2](#), respectively. Note that the number of anomalies in the fourth column refers to the number of occurrences of anomalies in each feature, not in each flight. Thus, for the same flight, anomalies may occur in multiple (possibly all) features and also multiple times in a single feature.

The following observations can be made from [Tables 1 and 2](#):

- For the departures, LGA and JFK generally have a higher number of anomalies per flight compared to EWR, which could be explained by more stringent navigation requirements within the shared airspace of LGA and JFK due to the proximity of both airports to each other (~ 7.3 nm), as shown in [Fig. 3](#).
- The frequency of anomalies in the vertical and speed features per flight is significantly lower for departures than for arrivals. This is due to the characteristics of terminal airspace operations, where typically, policies require only a lower bound or minimum for the speed or altitude of departing flights, and a maximum vertical speed climb is permitted, while more stringent requirements are applied to arriving flights.

We present illustrative anomaly detection models (in blue) generated for each of the five features in [Fig. 10](#). The normal flights are shown in green while the anomalous flights are shown in red:

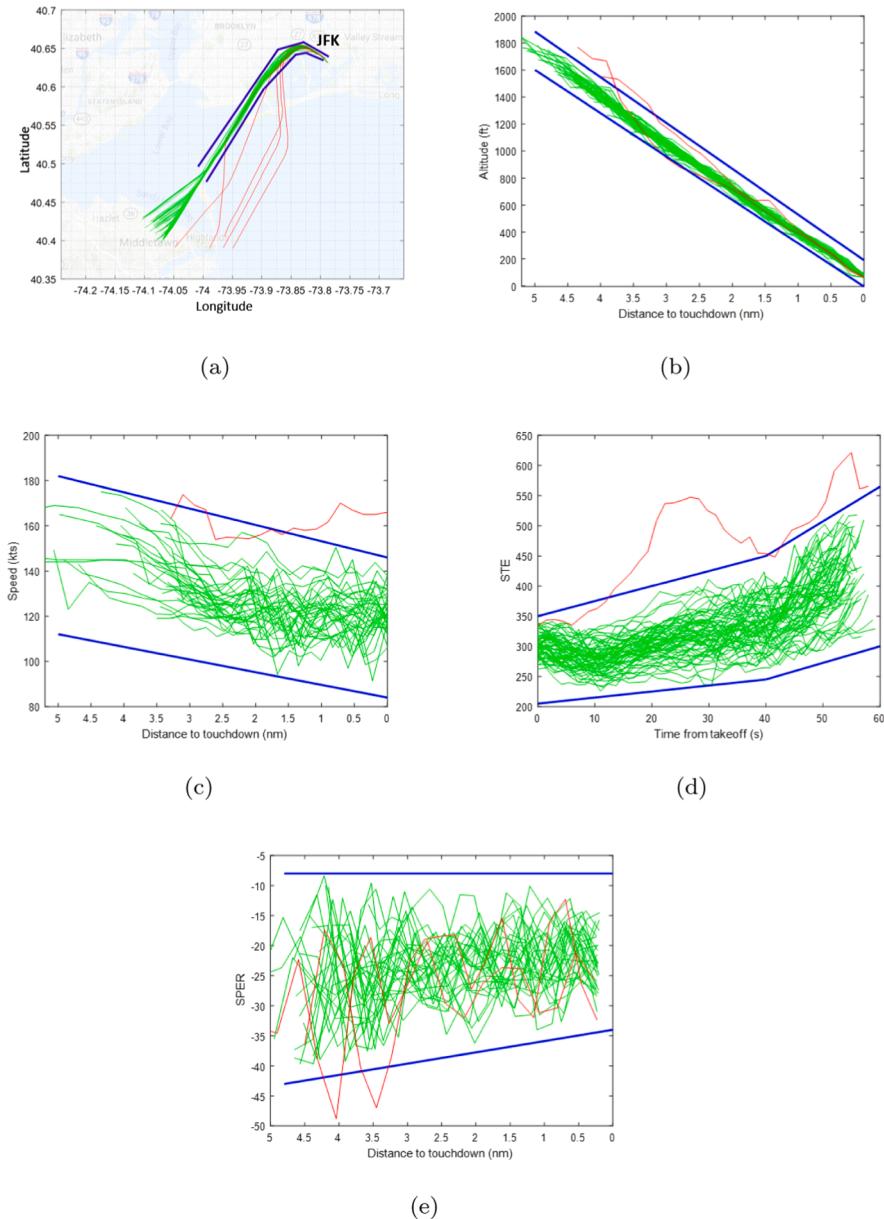


Fig. 10. Illustrative anomaly detection models for individual features: (a) Horizontal anomaly detection model for departures from JFK RWY31R; (b) Vertical anomaly detection model for arrivals to EWR RWY 4L; (c) Speed anomaly detection model for arrivals to LGA RWY22; (d) Specific total energy anomaly detection model for departures from LGA RWY22; and, (e) Specific potential energy rate anomaly detection model for arrivals to JFK RWY31R.

- **Horizontal:** Fig. 10(a) shows the horizontal anomaly detection model for departing flights, where the flights not following their designated climb path toward the south departure exit gate after takeoff are detected as anomalies.
- **Vertical:** Fig. 10(b) shows the vertical anomaly detection model for arriving flights, where the flights above the nominal approach glide slope are detected as anomalies.
- **Speed:** Fig. 10(c) shows the speed anomaly detection model for arriving flights, where the flight landing at a higher-than-normal approach speed is detected as an anomaly.
- **STE:** Fig. 10(d) shows the STE anomaly detection model for departing flights, where the flight with excess STE is detected as an anomaly.
- **SPER:** Fig. 10(e) shows the SPER anomaly detection model for arriving flights, where the flights with a high sink rate while landing are detected as anomalies.

Table 3

Distribution of horizontal anomalies for departures, detected using event-triggered TempAD and TempAD-OU.

Airport	No. of flights	No. of anomalies		Anomalies per 1000 departure flights	
		Event-triggered TempAD	TempAD-OU	Event-triggered TempAD	TempAD-OU
LGA	37,112	464	128	12.5	3.4
JFK	41,391	519	301	12.3	7.2
EWR	43,778	496	294	11.2	6.7

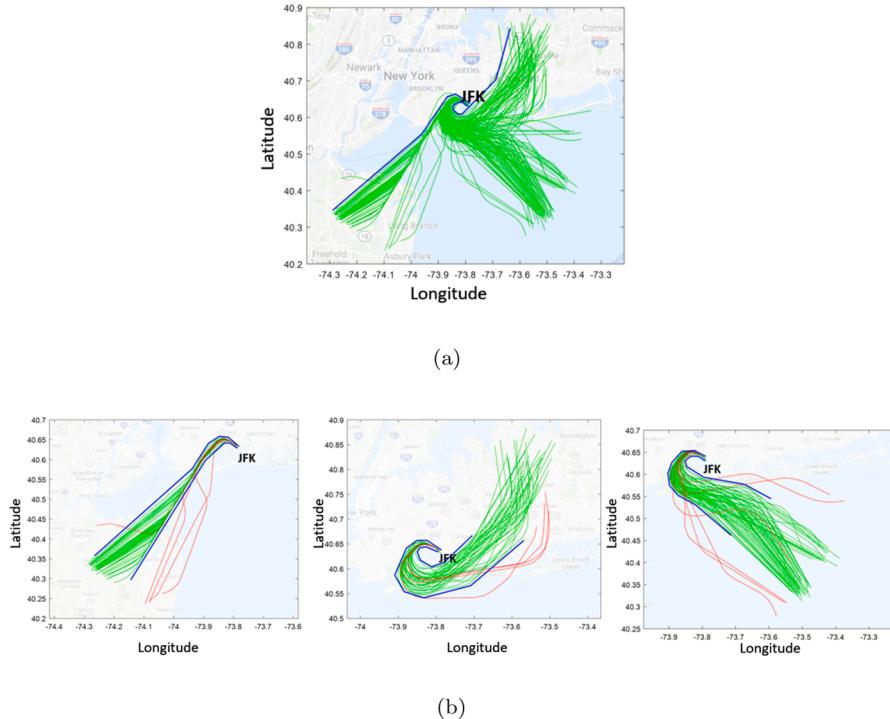


Fig. 11. Horizontal anomaly detection models for departures from JFK RWY31L during an entire day: (a) Model generated using TempAD-OU; and, (b) Models generated using event-triggered TempAD (southwest gate, east gate and water gate, respectively).

3.2. Comparison of anomaly detection performance

We compare the distribution of anomalies detected by the proposed algorithm against the TempAD-OU algorithm (Deshmukh and Hwang, 2019) that we developed. This TempAD-OU algorithm is an incremental learning anomaly detection algorithm as well, but the model library in this case is periodically updated every 24 h (overnight). In addition, the algorithm lacks the input of the metroplex operations dataset and relies only on the surveillance dataset to maintain and update the model library. For illustration, comparing only the horizontal feature, Table 3 presents the results for the horizontal anomalies detected by event-triggered TempAD and TempAD-OU.

We observe that anomaly detection capability is significantly improved by using the airport operations data in event-triggered TempAD. To validate that these anomalies are in fact true operational anomalies, some illustrative and comparative results of event-triggered TempAD against TempAD-OU are presented next. In the context of the results presented in the next subsection, operational anomalies are those flights which do not conform to the operational policies, such as flights that do not follow the mandated climb path.

3.3. Illustrative study of anomaly detection models

The operational events concerning some airports can be ‘self-reliant’, i.e., they are typically dependent only on operational parameters recorded at that airport. On the other hand, some airports require operational parameters recorded in the entire metroplex. These two cases have been demonstrated separately.

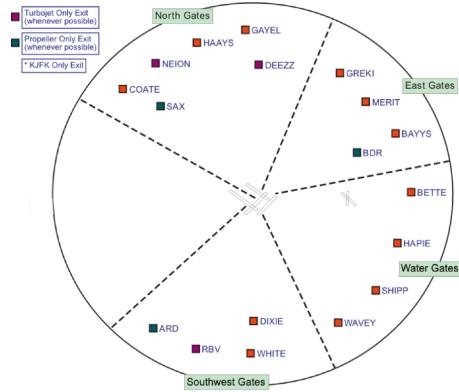


Fig. 12. JFK departure gates.

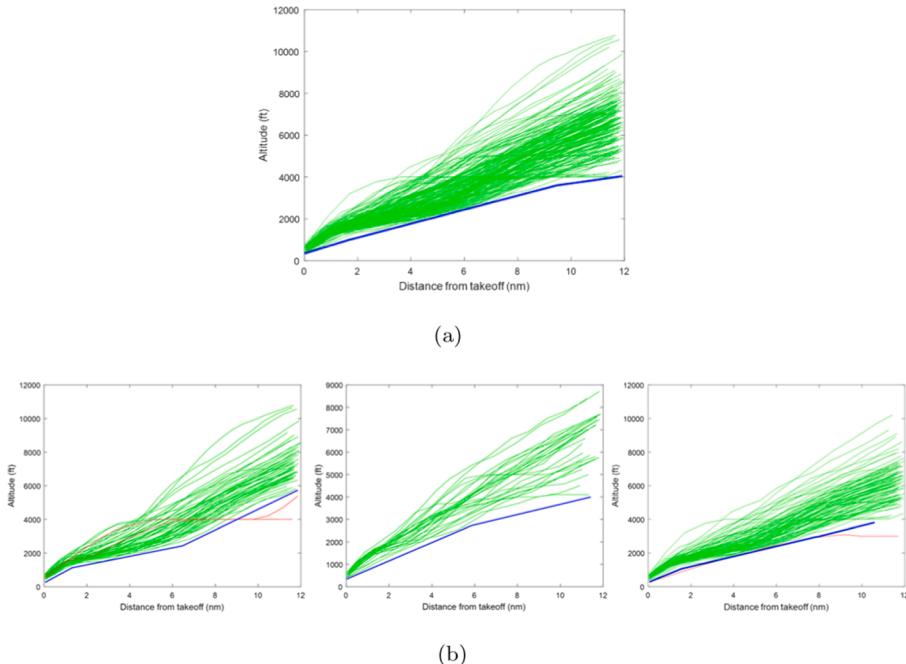


Fig. 13. Vertical anomaly detection models for departures from JFK RWY 31L during an entire day: (a) Model generated using TempAD-OU, and, (b) Models generated using event-triggered TempAD (southwest gate, east gate and water gate, respectively).

3.3.1. Single airport case: JFK

Operations at JFK typically are unaffected by LGA or EWR operations. Thus, operational policies at JFK are mandated only based on incremental parameters recorded at JFK.

Horizontal anomaly detection. Fig. 11(a) presents the horizontal anomaly detection model generated by TempAD-OU for all departures from runway 31L at JFK. We can clearly see the ‘fanning’ of the trajectories (in green) and that the model (in blue) generated for this entire data recording is too conservative and thus could miss several operational anomalies.

Upon using event-triggered TempAD to detect anomalies (in red) within the same dataset, the framework presented in Fig. 6 relies on the departure exit gate feature to generate more effective anomaly detection models, as shown in Fig. 11(b). The departure exit gates mandated by FAA (New York ARTCC - JFK SOP, 2019) are presented in Fig. 12. We observe that the anomalies here correspond to the trajectories that do not follow a designated climb path.

Vertical anomaly detection. Similarly, we compare the vertical anomaly detection models for departures from runway 31L during the same day. Figs. 13(a) and (b) present the vertical anomaly detection models generated for departures using TempAD-OU and event-triggered TempAD, respectively.

To demonstrate the effectiveness of the anomaly detection models generated by the proposed algorithm, we observe how well the anomaly detection models mimic the policies mentioned in the FAA documents. For the east gate departures, operational policies

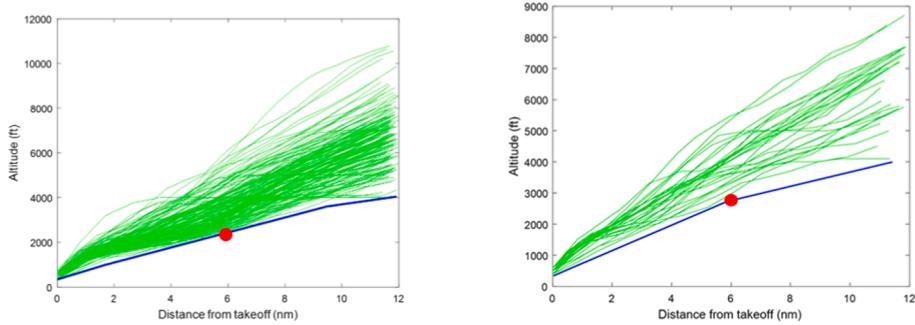


Fig. 14. Comparison of minimum altitudes discovered by TempAD-OU (left) and event-triggered TempAD (right).

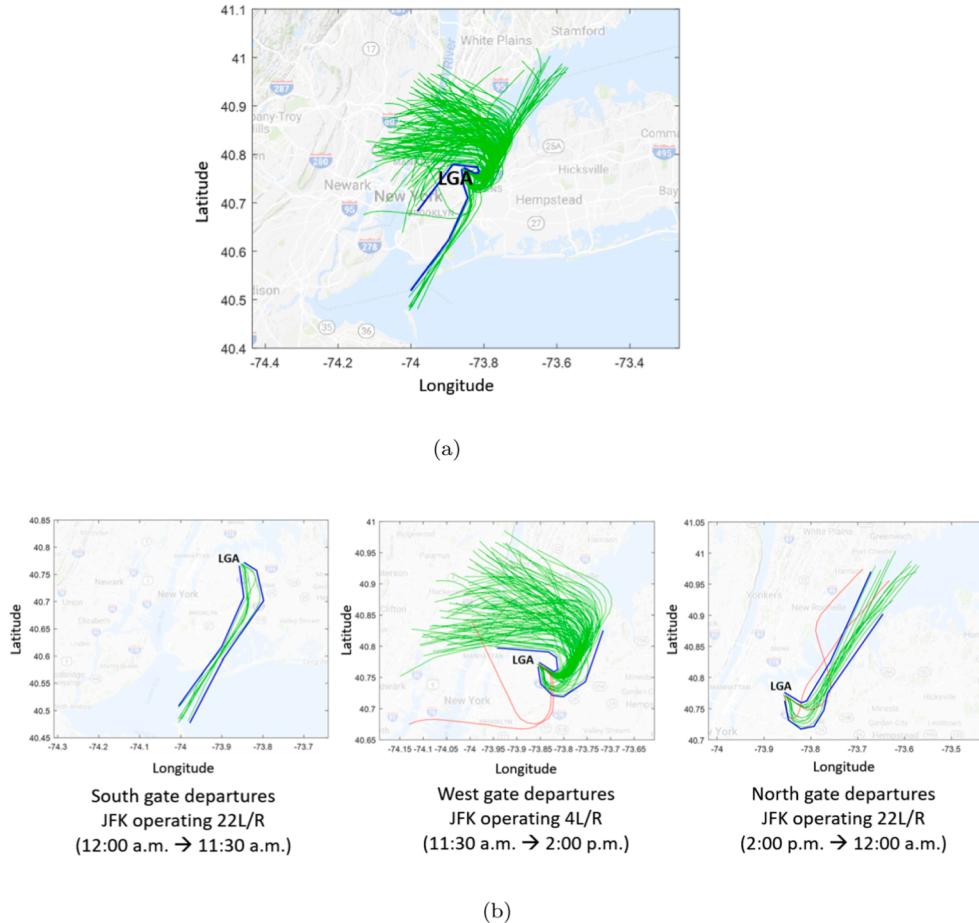


Fig. 15. Horizontal anomaly detection models for departures from LGA RWY13 during an entire day: (a) Model generated using TempAD-OU; and, (b) Models generated using event-triggered TempAD (south gate, west gate and north gate, respectively).

mandate a minimum altitude of 2,500 feet at 6 nautical miles from takeoff. The TempAD-OU model for the same departures shows a lower bound of 2,000 feet, which violates the policy, while the enhanced event-triggered TempAD model shows a lower bound of 2,700 feet, as shown by the red solid circles in Fig. 14. Thus, the safety bound given by operational policies is successfully captured using the event-triggered TempAD algorithm.

3.3.2. Multi-airport case: LGA and EWR

In contrast, the terminal airspace operations at LGA and EWR are significantly affected by operations at the other two airports, and features relating to more complex multi-airport coordination strategies have to be included to generate effective models. For example, the operating runway at JFK affects the climb path used for departures from LGA. The operating runway changes only a few times at

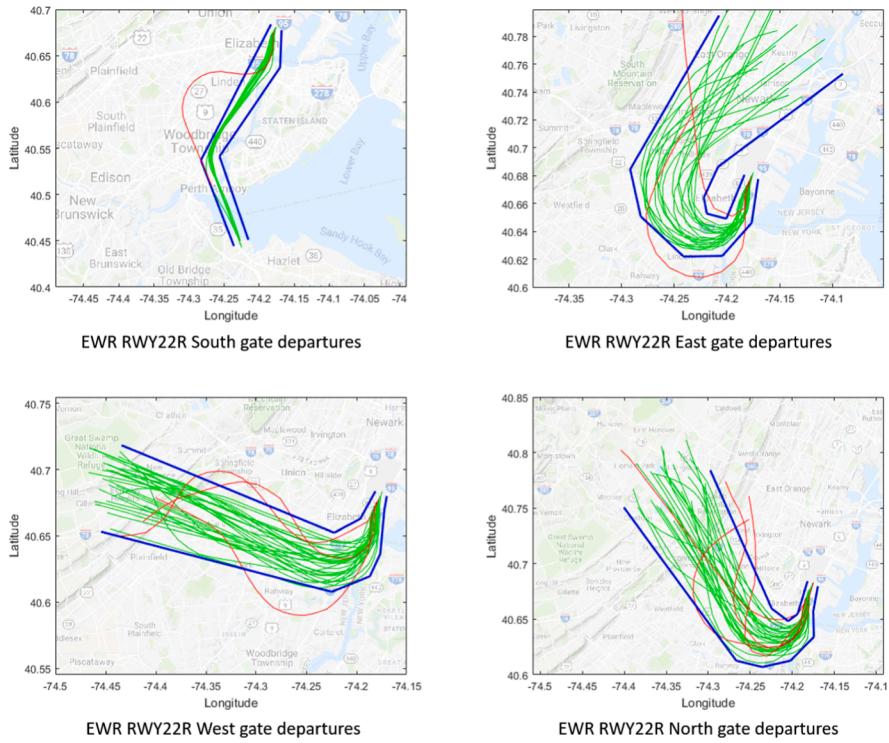


Fig. 16. Horizontal anomaly detection models for departures from EWR RWY22R, generated using event-triggered TempAD.

JFK throughout the day. In the illustration presented next, the JFK operating runway is 22L/R between midnight and 11:30 a.m., 4L/R between 11:30 a.m. to 2:00 p.m., and 22L/R subsequently till midnight. Fig. 15(a) presents the horizontal anomaly detection model generated for departures from LGA runway 13, in the absence of airport operations data (using TempAD-OU).

Upon providing features relating to the airport operations data as an input, specifically the operating runway at JFK and the multi-airport coordination policies from (New York ARTCC - LGA SOP, 2019), we observe that the departure models segregate into the south, west and north gate departures, as shown in Fig. 15(b), and are capable of detecting horizontal operational anomalies more effectively. A notable observation in this figure is that it demonstrates the advantage of the proposed algorithm over typical trajectory clustering-based algorithms. The proposed algorithm can more effectively capture the dispersed end parts of departure trajectories before the flights leave the metroplex airspace; on the other hand, a typical clustering-based anomaly detection algorithm would generate a number of spurious anomalies due to high variance in this region.

A similar analysis can be performed for EWR operations as well. Fig. 16 illustrates the horizontal anomaly detection models generated for all departures from runway 22R at EWR during an entire day, wherein multiple models are generated using the departure gates feature in the event-triggered TempAD algorithm.

From these analyses and illustrations, we observe that the event-triggered TempAD algorithm is capable of adapting to the evolving metroplex operations and thus generate anomaly detection models that can effectively detect operational anomalies.

4. Conclusion

The main contribution of this paper is to develop an event-based incremental learning algorithm for anomaly detection in metroplex operations, which is capable of ingesting both air traffic surveillance and airport operations datasets. Existing anomaly detection algorithms rely only on surveillance data as an input, thereby losing vital information present in the operations dataset, such as operating runway, multi-airport coordination strategies, and departure exit gates. The proposed algorithm uses both these datasets together to obtain more effective anomaly detection models, as the generated models now take into consideration the effect of co-ordinated operations on the arrival and departing flights in the metroplex airspace. The proposed algorithm, called event-triggered TempAD, has been demonstrated using real air traffic surveillance and operations datasets for all operations in the New York metroplex comprising the LaGuardia, John F. Kennedy, and Newark airports. The analysis of results using this algorithm showed that it outperforms algorithms that rely only on surveillance data as input: qualitatively – by generating anomaly detection models that more closely resemble the operational evolution of the metroplex airspace; and quantitatively – by improving the effectiveness of capturing true operational and safety-related anomalies, as well as detecting anomalies sooner.

In the future, we plan to use the proposed event-triggered TempAD in a real-time setting, whereby it can detect anomalies in streaming data. The resulting tool can be used to aid pilots and air traffic controllers in real-time decision making to avoid the

occurrence of anomalies, thereby improving the throughput performance and safety in the terminal airspace.

CRediT authorship contribution statement

Raj Deshmukh: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Dawei Sun:** Validation, Investigation, Data curation. **Kwangyeon Kim:** Conceptualization, Writing - review & editing. **Inseok Hwang:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgements

This research is funded by the National Aeronautics and Space Administration (NASA) through the NASA Big Data Analytics project (NNH15ZEA001N). The authors are grateful to Dr. Andrew Churchill at Mosaic ATM, Megan Hawley at Honeywell, and Dr. Nikunj C. Oza at NASA Ames for their valuable comments and continued support.

References

- A.C.A.S. Authority, 2011. Guidance on the establishment of a flight data analysis program–safety management systems (CAAP SMS-4 (0)), Australian: Civil Aviation Advisory Publication.
- Agrawal, S., Agrawal, J., 2015. Survey on anomaly detection using data mining techniques. *Procedia Comput. Sci.* 60, 708–713.
- Aviation System Performance Metrics (ASPM) Web Data System, May 2019. URL [https://aspmhelp.faa.gov/index.php/Aviation_Performance_Metrics_\(APM\)](https://aspmhelp.faa.gov/index.php/Aviation_Performance_Metrics_(APM)).
- Clarke, J.P., Ren, L., McClain, E., Schleicher, D., Timar, S., Saraf, A., Crisp, D., Gutterud, R., Laroza, R., Thompson, T., et al., 2012. Evaluating concepts for operations in metroplex terminal area airspace. *J. Aircraft* 49 (3), 758–773.
- Cortes, C., Vapnik, V., 1995. Support-vector networks. *Machine Learning* 20 (3), 273–297.
- Dani, M.C., Freixo, C., Jollois, F.-X., Nadif, M., 2015. Unsupervised anomaly detection for aircraft condition monitoring system. In: Aerospace Conference, 2015 IEEE, IEEE, pp. 1–7.
- Das, S., Matthews, B.L., Srivastava, A.N., Oza, N.G., 2010. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 47–56.
- Das, S., Matthews, B.L., Lawrence, R., 2011. Fleet level anomaly detection of aviation safety data. In: Prognostics and Health Management (PHM), 2011 IEEE conference on, IEEE, pp. 1–10.
- Deshmukh, R., Hwang, I., 2019. Incremental-learning-based unsupervised anomaly detection algorithm for terminal airspace operations. *J. Aerospace Informat. Syst.* 16 (9), 362–384.
- Deshmukh, R., Hwang, I., 2019. Anomaly Detection Using Temporal Logic Based Learning for Terminal Airspace Operations. In: AIAA Scitech 2019 Forum, p. 0682.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231.
- FAA, A.C., 2017. 120–71B, Standard Operating Procedures and Pilot Monitoring Duties for Flight Deck Crewmembers.
- Fainekos, G.E., Pappas, G.J., 2009. Robustness of temporal logic specifications for continuous-time signals. *Theoret. Comput. Sci.* 410 (42), 4262–4291.
- Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. *Machine Learning* 29 (2–3), 131–163.
- Gallego, C.E.V., Comendador, V.F.G., Carmona, M.A.A., Valdés, R.M.A., Nieto, F.J.S., Martínez, M.G., 2019. A machine learning approach to air traffic interdependency modelling and its application to trajectory prediction. *Transport. Res. Part C: Emerg. Technol.* 107, 356–386.
- Gavrilovski, A., Jimenez, H., Mavris, D.N., Rao, A.H., Shin, S., Hwang, I., Marais, K., 2016. Challenges and opportunities in flight data mining: A review of the state of the art. In: AIAA Infotech@ Aerospace, p. 0923.
- Ghorbani, A.A., Lu, W., Tavallaei, M., 2009. Network intrusion detection and prevention: concepts and techniques, vol. 47. Springer Science & Business Media.
- Gorinevsky, D., Matthews, B., Martin, R., 2012. Aircraft anomaly detection using performance models trained on fleet data. In: *2012 Conference on Intelligent Data Understanding (CIDU)*. IEEE, pp. 17–23.
- Hodge, V., Austin, J., 2004. A survey of outlier detection methodologies. *Artificial Intell. Rev.* 22 (2), 85–126.
- IATA, 2019. Airlines' Perspective on Air Traffic Management. <http://www.atmseminar.org/seminarContent/seminar13/ATM2019KeynoteAlexisvonHoensbroech.pdf>
- Jones, A., Kong, Z., Belta, C., 2014. Anomaly detection in cyber-physical systems: A formal methods approach. In: *53rd IEEE Conference on Decision and Control*. IEEE, pp. 848–853.
- Kim, K., Hwang, I., 2018. Terminal Airspace Anomaly Detection Using Temporal Logic Learning. In: *ICRAT 2018 Third International Conference on Research in Air Transportation*, ICRAT Barcelona, Spain.
- Kong, Z., Jones, A., Medina Ayala, A., Aydin Gol, E., Belta, C., 2014. Temporal logic inference for classification and prediction from data. In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*. ACM, pp. 273–282.
- Kong, Z., Jones, A., Belta, C., 2017. Temporal logics for learning and detection of anomalous behavior. *IEEE Trans. Autom. Control* 62 (3), 1210–1222.
- Li, L., Gariel, M., Hansman, R.J., Palacios, R., 2011. Anomaly detection in onboard-recorded flight data using cluster analysis. In: *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. IEEE, pp. 4A4–1.
- Li, L., Das, S., Johns Hansman, R., Palacios, R., Srivastava, A.N., 2015. Analysis of flight data using clustering techniques for detecting abnormal operations. *J. Aerospace Informat. Syst.* 12 (9), 587–598.
- Li, L., Hansman, R.J., Palacios, R., Welsch, R., 2016. Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring. *Transport. Res. Part C: Emerg. Technol.* 64, 45–57.
- Ma, J., Perkins, S., 2003. Time-series novelty detection using one-class support vector machines. In: *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3, IEEE, pp. 1741–1745.
- Modernization of U.S. Airspace: Nextgen, Sept 2019. URL <https://www.faa.gov/nextgen/>.
- Narasimhan, S., Brownston, L., 2007. HyDE-A general framework for stochastic and hybrid model-based diagnosis. *Proc. DX* 7, 162–169.
- New York ARTCC – JFK SOP, Mar 2019. https://nyartcc.org/znywiki/index.php/JFK_SOP.
- New York ARTCC – LGA SOP, Mar 2019. https://nyartcc.org/znywiki/index.php/LGA_SOP.
- Oster, C.V., Oster, C.V., Strong, J.S., Zorn, C.K., Zorn, C.K., 1992. Why airplanes crash: Aviation safety in a changing world. Oxford University Press on Demand.
- Puranik, T.G., Mavris, D.N., 2017. Anomaly detection in general-aviation operations using energy metrics and flight-data records. *J. Aerospace Informat. Syst.* 1–14.
- Schwabacher, M., Oza, N., Matthews, B., 2009. Unsupervised anomaly detection for liquid-fueled rocket propulsion health monitoring. *J. Aerospace Comput. Informat. Commun.* 6 (7), 464–482.
- Shin, S., Hwang, I., 2017. Data-mining-based computer vision analytics for automated helicopter flight state inference. *J. Aerospace Informat. Syst.* 652–662.
- SMART-NAS for Safe Trajectory Based Operations Project Description, Aug 2017. <https://www.nasa.gov/aeroresearch/programs/aosp/smart-nas-project-description/>.

- Srivastava, A.N., 2012. Greener aviation with virtual sensors: a case study. *Data Min. Knowl. Disc.* 24 (2), 443–471.
- SWIM Terminal Data Distribution System (STDDS), Oct 2017. https://www.faa.gov/air_traffic/technology/swim/stdds/media/FIXM_Mediated_STDDS_Data_Overview_v2.pdf.
- Vaidya, A., Lee, S., Hwang, I., 2016. Data-driven modeling and analysis framework for cockpit human-machine interaction issues. *J. Aerospace Informat. Syst.* 370–380.
- Wan, E.A., 1990. Neural network classification: A Bayesian interpretation. *IEEE Trans. Neural Networks* 1 (4), 303–305.
- Xue, M., Zelinski, S., 2014. Dynamic stochastic scheduler for integrated arrivals and departures. In: 2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC). IEEE, pp. 1A2–1.
- Zhang, M.-L., Zhou, Z.-H., 2005. A k-nearest neighbor based algorithm for multi-label classification. *GrC* 5, 718–721.
- Zhu, S.-P., Huang, H.-Z., Peng, W., Wang, H.-K., Mahadevan, S., 2016. Probabilistic physics of failure-based framework for fatigue life prediction of aircraft gas turbine discs under uncertainty. *Reliab. Eng. Syst. Saf.* 146, 1–12.