DATA-DRIVEN ANOMALY AND PRECURSOR DETECTION IN METROPLEX AIRSPACE OPERATIONS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Raj Deshmukh

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Inseok Hwang, Chair

School of Aeronautics and Astronautics

Dr. Jianghai Hu

School of Electrical and Computer Engineering

Dr. Shaoshuai Mou

School of Aeronautics and Astronautics

Dr. Dengfeng Sun

School of Aeronautics and Astronautics

Approved by:

Dr. William Crossley

Head of the Departmental Graduate Program

Dedicated to my parents, Priya and Kiran Deshmukh

ACKNOWLEDGMENTS

Completing this dissertation means a lot to me, and could not have been possible without the help and guidance of several people.

First and foremost, I would like to thank my advisor, Prof. Inseok Hwang, for his constructive guidance and direction throughout the last five years and for helping me devise and develop ideas for machine learning applications in ATM and distributed state estimation. His insight, persistence, and enthusiasm have been critical in making the researcher and engineer I am. I am also grateful to my other committee members – Prof. Jianghai Hu, Prof. Shaoshuai Mou and Prof. Dengfeng Sun – for their time and inputs on my research. I am also 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.

I want to express my special gratitude to my mentors during all stages of my research: Prof. Cheolhyeon Kwon and Kwangyeon Kim. I savor the countless hours that were spent slaving away together discussing the complex math & symbolism, the results and the implications of my research with my labmates. I thank Aisorya and Dee for existing, and for their constant vital inputs to my life. I am indebted to Radhiks :), Jaibai, Omanshoe and Ankit Preeti Deo for making my time at Purdue enjoyable. Notch, unsheathe, and roll for initiative! Lastly, I thank my parents and my family for their everlasting love, encouragement, and tolerance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Deshmukh, Raj Ph.D., Purdue University, May 2020. Data-Driven Anomaly and Precursor Detection in Metroplex Airspace Operations. Major Professor: Inseok Hwang.

The air traffic system is one of the most complex and safety-critical systems, which is expected to grow at an average rate of 0.9% a year – from 51.8 million operational activities in 2018 to 62 million in 2039 – within the National Airspace System. In such systems, it is important to identify degradations in system performance, especially in terms of safety and efficiency. Among the operations of various subsystems of the air traffic system, the arrival and departure operations in the terminal airspace require more attention because of its higher impact (about 75% incidents) on the entire system's safety, ranging from single aircraft incidents to multi-airport congestion incidents.

The first goal of this dissertation is to identify the air traffic system's degradations – called anomalies – in the multi-airport terminal airspace or metroplex airspace, by developing anomaly detection models that can separate anomalous flights from normal ones. Within the metroplex airspace, airport operational parameters such as runway configuration and coordination between proximal airports are a major driving factor in aircraft's behaviors. As a substantial amount of data is continually recording such behaviors through sensing technologies and data collection capabilities, modern machine learning techniques provide powerful tools for the identification of anomalous flights in the metroplex airspace. The proposed algorithm ingests heterogeneous data, comprising the surveillance dataset, which represents an aircraft's physical behaviors, and the airport operations dataset, which reflects operational procedures at airports. Typically, such aviation data is unlabeled, and thus the proposed algorithm is developed based on hierarchical unsupervised learning approaches for anomaly detection.

This base algorithm has been extended to an anomaly monitoring algorithm that uses the developed anomaly detection models to detect anomalous flights within real-time streaming data.

A natural next-step after detecting anomalies is to determine the causes for these anomalies. This involves identifying the occurrence of precursors, which are triggers or conditions that precede an anomaly and have some operational correlation to the occurrence of the anomaly. A precursor detection algorithm is developed which learns the causes for the detected anomalies using supervised learning approaches. If detected, the precursor could be used to trigger actions to avoid the anomaly from ever occurring.

All proposed algorithms are demonstrated with real air traffic surveillance and operations datasets, comprising of departure and arrival operations at LaGuardia Airport, John F. Kennedy International Airport, and Newark Liberty International Airport, thereby detecting and predicting anomalies for all airborne operations in the terminal airspace within the New York metroplex. Critical insight regarding air traffic management is gained from visualizations and analysis of the results of these extensive tests, which show that the proposed algorithms have a potential to be used as decision-support tools that can aid pilots and air traffic controllers to mitigate anomalies from ever occurring, thus improving the safety and efficiency of metroplex airspace operations.

1. INTRODUCTION

This chapter presents a detailed literature survey of anomaly and precursor detection algorithms along with the motivation that propels this research of such algorithms applicable to air traffic management (ATM). This is followed by a post-hoc summary of the contributions of this dissertation.

1.1 Background and motivation

1.1.1 Anomaly detection

The ATM system is one of the most complex and continuously-evolving systems, where a number of pilots and air traffic controllers (ATCs) coordinate and control navigation of flights. The global air traffic density doubles roughly every 15 years [1], 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) [2] and SMART-NAS (Shadow Mode Assessment Using Realistic Technologies for the National Airspace System) [3] 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 higher impact on the entire system's safety (about 75% of incidents in the NAS) [4, 5]. Of these, the terminal airspace of a metropolitan area where multiple closely-located airports coordinate with each other, called *metroplex* [6], such as in New York, Houston, Denver, and Las Vegas, is key in ensuring and enhancing the safety of 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 as conflicts between aircraft (loss of separation) or issues for a single aircraft (e.g., proximity to the ground, passenger discomfort, fatigue for the airframe, etc.) [7, 8]. The objective 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 [9] and the dynamics of an aircraft [10]; 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 [11, 12]. 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 [13], support vector machines (SVMs) [14], k-nearest neighbors [15], Bayesian networks [16], and decision trees [17]. In aviation applications, regression-based supervised learning algorithms have been employed to model aerodynamic force data [18] and to detect anomalies in engine fuel consumption [19]. 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 [20]. 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 [21–24]. In [25], 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 [7] 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 [26] and in routine airline operations [27]. The one-class support vector machine (OCSVM) based methods attempt to design a hyperplane that has all normal data on one side [28] and have been applied to anomaly detection in propulsion health monitoring [29] and in energy metrics for general aviation operations [30].

1.1.2 Precursor detection

Once anomalies are detected, a natural extension is to find the causes for these anomalies, in order to aid ATCs and pilots in mitigating or avoiding such anomalous behaviors to ensure safe and efficient operations in the airspace. The approach to finding the causes – called *prognosis* – requires the identification of events or conditions that precede the anomaly with some correlation to the occurrence of the anomaly. The algorithms that detect those events or conditions – called *precursors* – can be served as a decision support tool for ATCs and pilots.

There have been extensive research efforts to develop precursor detection algorithms in aviation applications. One approach is to develop a physics-based or a rule-based model of the system, which relies on models designed by domain experts through a thorough analysis of the dynamics of the system [29]. Common examples of such an approach is system health monitoring [31, 32] and conformance monitoring [33, 34]. Another approach is the data-driven approach that uses large records of historical data to learn a model of system behavior. In the aviation domain, data-driven approaches have been applied to structural, gas turbine, and battery prognostics [35–37].

The airspace is a very complex system with a continuously evolving behavior, making it challenging to determine its governing equations, and thus, a physics-based or rule-based analysis for precursor detection in the airspace is typically not feasible. A challenge to the data-driven precursor detection approach is the availability of anomalous data for training. Within the air traffic management system, the terminal airspace has the highest traffic density, and thus more anomalies are expected. Therefore, the problem of precursor detection in air traffic management is best addressed by applying machine learning techniques to surveillance data obtained in the terminal airspace.

In [38], the authors have proposed a reinforcement learning-based algorithm called Automatic Discovery of Precursors in Time series data (ADOPT), which generates a model that separates the normal time-series flight operations quality assurance (FOQA) data from the adverse (or anomalous) FOQA data. The authors have demonstrated ADOPT to detect precursors to go-around anomalies by searching for suboptimal actions in the adverse time series, which are actions that increase the risk of an abnormal event, by deviating from the sequence of optimal actions. This sequence of optimal actions is determined by ADOPT as a Markov decision process corresponding to normal flight operations. The ADOPT algorithm is capable of searching for precursors within only those physical states (such as groundspeed, headwind, closing speed to runway, etc.) for which the reinforcement learning model is trained. This could limit the applicability of the algorithm for precursor detection, as the precursor may occur in a dimension not corresponding to the physical states, but rather in a complex combination of multiple states which may not necessarily have any intuitive interpretation.

1.2 Objectives and contributions

1.2.1 Anomaly detection

There are certain desirable properties that an unsupervised learning-based anomaly detection algorithm which can detect anomalous flights from aviation data should have:

- Effectiveness: The algorithm should be able to discern anomalies from normal data effectively, i.e., the learned anomaly detection model should consider properties such as tightness of data, and the separation between anomalous and normal data.

- Operations-aware: Considering that the terminal airspace – especially metroplex airspace – is highly structured and strategically-controlled according to policies, the algorithm must take into account the impact of operations on normal behaviors in the airspace. Most existing 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. Thus, the algorithm's input should be heterogeneous, comprising of both surveillance data and airport operations data.

- Human-interpretable: Human feedback to improve the anomaly monitoring algorithms is critical in reducing the number of false alarms and ensuring that all true operational anomalies are detected during real-world operations.

Thus, the algorithm should generate anomaly detection models that are human-readable (natural language interpretable), making it easier to use by end-users (pilots/ATCs/policymakers), thereby allowing them to understand and analyze the results well and gain insight from the results of anomaly detection. Recently, research has been performed to design such an algorithm, where the output model is expressed as signal temporal logic (STL) bounds. The STL bounds are mathematical expressions in terms of temporal and state information, and are also human-readable and thus easily interpretable by domain experts [39–41].

- Incremental learning: Airspace and airport configuration evolve dynamically, often several times through a day. Thus, there is a requirement to maintain an anomaly detection model library from which the specific model applicable to the configuration in operation is chosen. Furthermore, newly recorded (incremental) data should be efficiently and recursively incorporated into the model library.

The above properties were developed sequentially, resulting in the research presented in this dissertation. A recursive temporal logic based anomaly detection algorithm – called event-triggered TempAD – is developed to generate models from mini-batch data, i.e., the models are updated iteratively after a data recording of a specific length is obtained. The proposed algorithm tackles the issues of computational complexity by processing smaller dataset sizes, instead of processing the entire dataset at once. More importantly, the generated models can capture anomalies more effectively as they adapt to variations in air traffic operations with every mini-batch update. With extensive testing and analysis of real air traffic data, it was established that this new proposed algorithm can effectively detect various types of aviation anomalies in air traffic operations in the terminal airspace. To demonstrate the potential of the algorithm to be used for real-time application, specific modifications in the framework of the algorithm are made to allow it to assimilate real-time flight data, followed by anomaly monitoring to warn pilots and ATCs.

1.2.2 Precursor detection

In this dissertation, an algorithm to detect precursors that correlate to specific anomalies in terminal airspace operations is proposed. This algorithm uses real air traffic surveillance data along with labels that indicate if a flight is anomalous as an input to determine precursors, which enables the use of supervised learning techniques to generate the precursor detection model. These models can detect precursors which may occur in extremely non-linear dimensions that are a complex combination of multiple physical states.

To generate the labels, the event-triggered TempAD algorithm is used to generate anomaly detection models that discern anomalies from normal data accurately. The proposed precursor detection algorithm – called *reactive TempAD* – is demonstrated with terminal airspace surveillance datasets, such as the Airport Surface Detection Equipment - Model X (ASDE-X) [42] and the Terminal Automation Information Service (TAIS) [43] datasets, which are obtained from aircraft during their final approach to the three airports in the New York metroplex.

1.3 Outline of dissertation

The rest of the dissertation is organized as follows: Chapter 2 presents the development and test of the proposed anomaly detection algorithm, which is thereafter demonstrated to effectively detect anomalies for flights within the terminal airspace of the New York metroplex. Illustrative analysis of these test results gives insight into the air traffic management system at this metroplex. Furthermore, an extension of this algorithm for real-time applications is also presented. In Chapter 3, the framework of the anomaly detection algorithm is modified to predict anomalies using supervised learning techniques. This precursor detection algorithm enables prognosis, and can potentially be used order to prevent the anomaly from even occurring. This algorithm is demonstrated to predict two specific anomaly types that are commonly observed in the terminal airspace as conflict mitigation strategies. Finally concluding discussions and potential future research to further enhance the anomaly and precursor detection capabilities are presented in Chapter 4.

2. ANOMALY DETECTION IN METROPLEX TERMINAL AIRSPACE OPERATIONS

This chapter presents the development of 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:

- individual aircraft's behavior following the procedures at a single airport (Fig. 2.1(a)); and
- collective aircraft behaviors according to the multi-airport coordination policies (Fig. 2.1(b) and 2.1(c)).

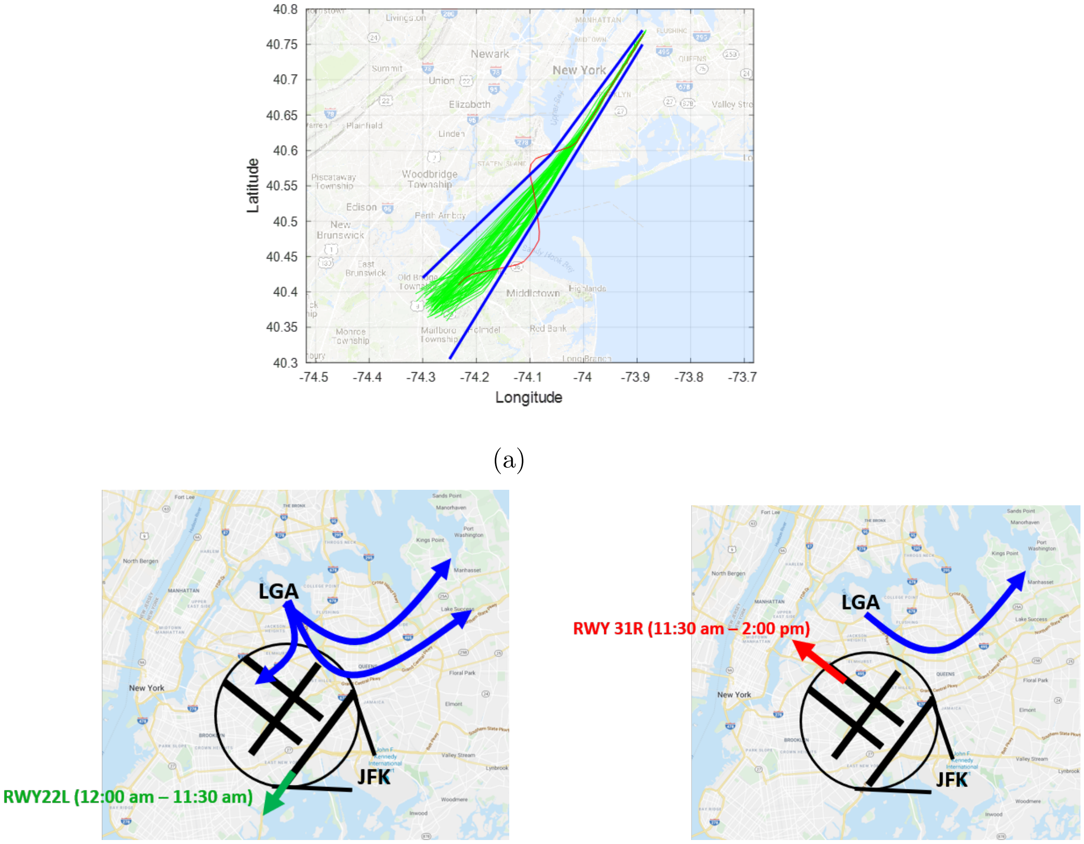
In this regard, the proposed algorithm ingests a heterogeneous input: (i) a surveillance dataset which records individual aircraft's states such as position and speed in a time-series format; and (ii) an operations dataset which contains information such as the operating runway, arrival and departure exit gates, and multi-airport coordination policies, which are discussed in Section 2.1.1.

2.1 Preliminaries

This section describes the inputs and data pre-processing required for effective application of the proposed algorithm.

2.1.1 Input data

The proposed algorithm takes as input two datasets: the surveillance dataset and operations dataset. The Terminal Automation Information Service (TAIS) [43]



(a)

(b)

(c)

Figure 2.1. 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.

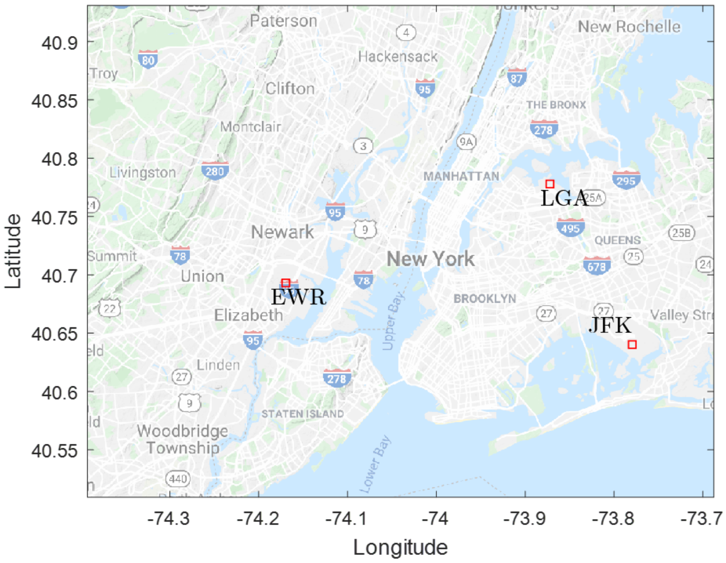


Figure 2.2. Major airports in the New York metroplex: LGA, JFK and EWR

dataset and the Aviation System Performance Metrics (ASPM) [44] dataset are used 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 seconds. 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 dissertation are recorded from September to November in 2016, in the New York metroplex (shown in Fig. 2.2), whose major airports are the LaGuardia airport (LGA), John F. Kennedy airport (JFK), and Newark airport (EWR).

2.1.2 Clustering

Structured operations in the terminal airspace mandate that approaches and climb paths have very distinct characteristics in terms of the allowed approach speeds, climb rate, heading, glide slope angles, etc. Thus, generating anomaly detection models on the entire dataset at once could yield very conservative results. These conservative models are ineffective, as they fail to detect some anomalous flights that have states buried within the entire dataset.

To circumvent the issue of conservativeness of models generated by using input data comprising of several approaches, a data pre-processing step is embedded that involves clustering of the source dataset using the popular DBSCAN (Density-based spatial clustering of applications with noise) algorithm [21]. For this purpose, DBSCAN is specifically chosen, as it is fast and has the ability to automatically determine the number of arbitrarily-shaped clusters and simultaneously detect outliers within time-series data, based on user-specified parameters [30], which correspond to the allowable minimum density of the clusters and to the minimum allowable population of a cluster. These parameters are chosen based on the physical density of each approach path and tuned until satisfactory clustering is observed. This clustering step ensures that all constituent time-series states within the same cluster have similar properties, making anomaly detection more effective and efficient. Then, the models for each of these clusters are generated separately.

Through observations from preliminary clustering, it was found that flight patterns are characteristics of the horizontal approach that flights take while landing at an airport, i.e., every flight taking the same approach to an airport has similar properties for the approach speed, approach glide slope angle, etc. This is further validated by the FAA documents [45–47] that dictate the approach procedures to airports. Thus, the clustering step is performed in the horizontal dimensions (aircraft positional data) and each cluster is then analyzed separately for every dimension. The result of the clustering step for arrivals to the LaGuardia (LGA) airport is shown in Fig. 2.3, where the figure on the left presents the 3-D trajectory of all arrivals during a day, while the figure on the right presents the centroids of the arrival patterns. The centroids are obtained for flights clustered based on their horizontal trajectories for the entire TAIS dataset between September to November 2016. There are 10 approach patterns identified for LGA (3 for runway 4, 3 for runway 22, 3 for runway 31 and 1 for

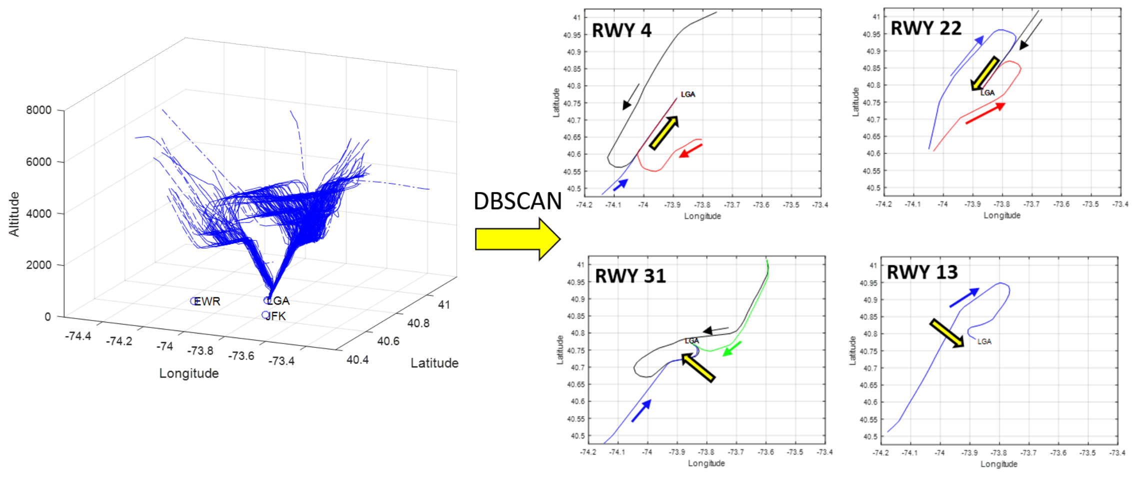


Figure 2.3. Clustering results for LGA arrivals

runway 22) during this period. Similarly, 10 approach patterns are identified for JFK and 13 runway patterns are identified during the same period. Note that similarly shaped approaches to parallel runways with the same alignment (such as runway 22L and 22R at JFK) are combined for analysis. Also, note that there may be more approach patterns possible than the ones observed within this specific TAIS dataset, as operations could change seasonally and new approaches may be used in other recording periods.

2.1.3 Feature selection

For each identified pattern, specific features corresponding to flight trajectories are selected to be used as input to the anomaly detection algorithm. The basic dimensions for anomaly detection are:

- Horizontal (H): Obtained from the positional (latitude and longitude) time-series data

- Vertical (V): Obtained from the altitude (h) time-series data

- Speed (S): Obtained from the ground speed (v) time-series data

In addition to these, a set of derived features are synthesized to represent the energy of the aircraft, calculated as shown in Table 2.1. Energy management is particularly important in the terminal airspace for detecting energy excess or deficient approaches [22,30]. Often, anomalies detected in the energy dimension are normal in the vertical and speed dimensions, as they are buried underneath the normal data.

Table 2.1. Energy Features

Feature	**Formula**
Specific Total Energy (STE)	$h + v^2/2g$
Specific Kinetic Energy (SKE)	$v^2/2g$
Specific Total Energy Rate (STER)	$\dot{h} + \dot{v}v/g$
Specific Kinetic Energy Rate (SKER)	$\dot{v}v/g$
Specific Potential Energy Rate (SPER)	$\dot{h}$

Here, g is the acceleration due to gravity (9.81 m/s^2). Among the energy features presented in Table 2.1, STE and SPER are chosen for anomaly detection, as they are most representative features which can capture properties of both altitude and speed effectively.

2.2 Algorithm development

In this section, each component within the framework of the proposed incremental learning based unsupervised anomaly detection algorithm is explained in detail.

2.2.1 Anomaly detection model: Temporal logic learning

Suppose there are N flights within a given pattern. The set of time-series of a feature are denoted as $\{s_i\}_{i=1}^{N}$ for the i-th flight. Since the time-series data is

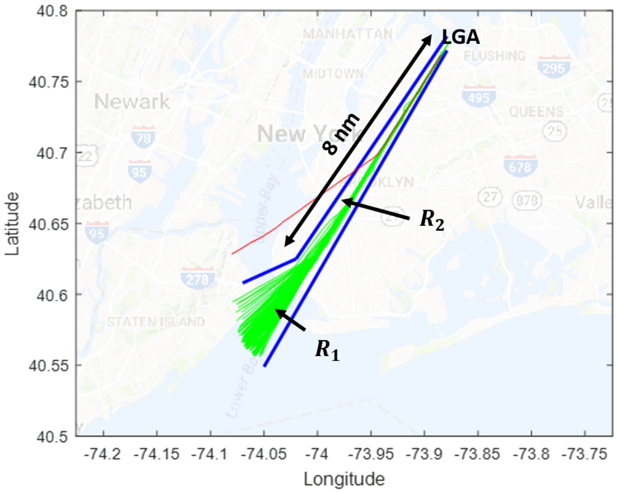


Figure 2.4. Horizontal anomaly detection model for arrivals to RWY4 at LGA

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. 2.4 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:

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

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:

$$R_1 = \{(x, y) : -0.411x + y - 71.0476 < 0 \text{ and } -1.3041x + y - 37.1171 > 0\} \; \forall \, t \in [10, 8]$$
$$R_2 = \{(x, y) : -1.1135x + y - 123.0444 < 0 \text{ and } -1.3041x + y - 37.1171 > 0\} \; \forall \, t \in [8, 0] \tag{2.1}$$

Expression (2.1) is referred to as a *temporal logic model* [41], 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 $ax + by + c < 0$).

2.2.2 Base model: TempAD algorithm

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. The model structure is determined 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, can be determined by optimizing the following cost function:

$$\min_{\theta,\epsilon} f\left(\{s_i\}_{i=1}^N, \varphi_\theta\right) = d\left(\{s_i\}_{i=1}^N, \varphi_\theta\right) + (-\epsilon) + \frac{1}{\alpha N} \sum_{i=1}^{N} \mu\left(s_i, \varphi_\theta\right) \tag{2.2}$$

Here,

- $d\left(\{s_i\}_{i=1}^N, \varphi_\theta\right)$ is a tightness function that penalizes the conservativeness of the model. Minimizing d prevents the learned model φ_θ from trivially describing all observed time-series data.

- ϵ is the gap between normal and anomalous time-series data. Maximizing ϵ makes the model robust to noise in the input data.

- $\mu\left(s_i, \varphi_\theta\right)$ 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 $(\epsilon/2)$.

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

  where $r(s_i, \varphi_\theta)$ is called the robustness degree, 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.

The robustness degree, $r(s_i, \varphi_\theta)$, 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. This framework of the algorithm used for this *base-learning algorithm* is presented in Fig. 2.5.

2.2.3 Incremental learning: Event-triggered TempAD

Next, the model library needs to be updated in order to incorporate the incremental data as well as keep the knowledge learned from the past dataset. For this purpose, an *event-based incremental learning algorithm* is proposed, 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. 2.6, which are presented in details in what follows.

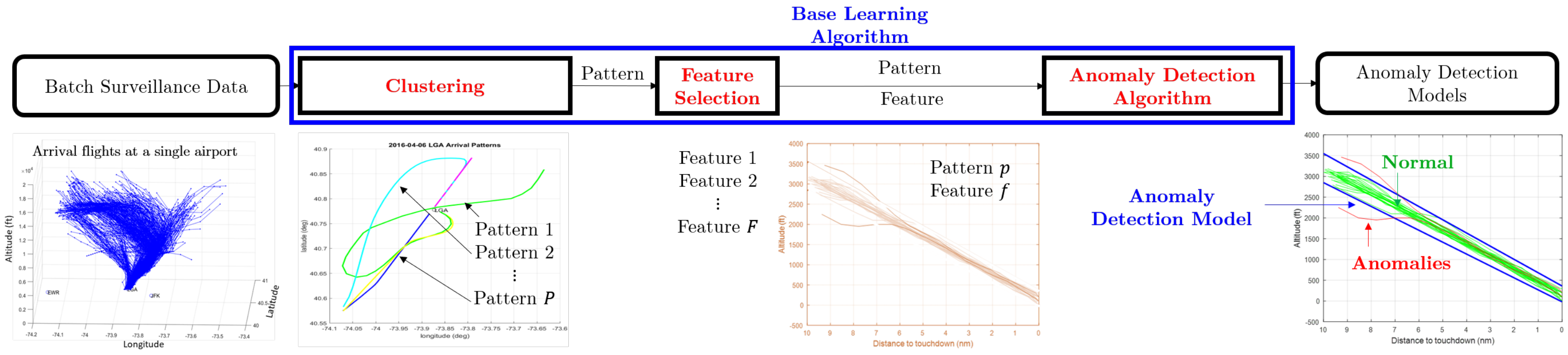


Figure 2.5. Base-learning algorithm

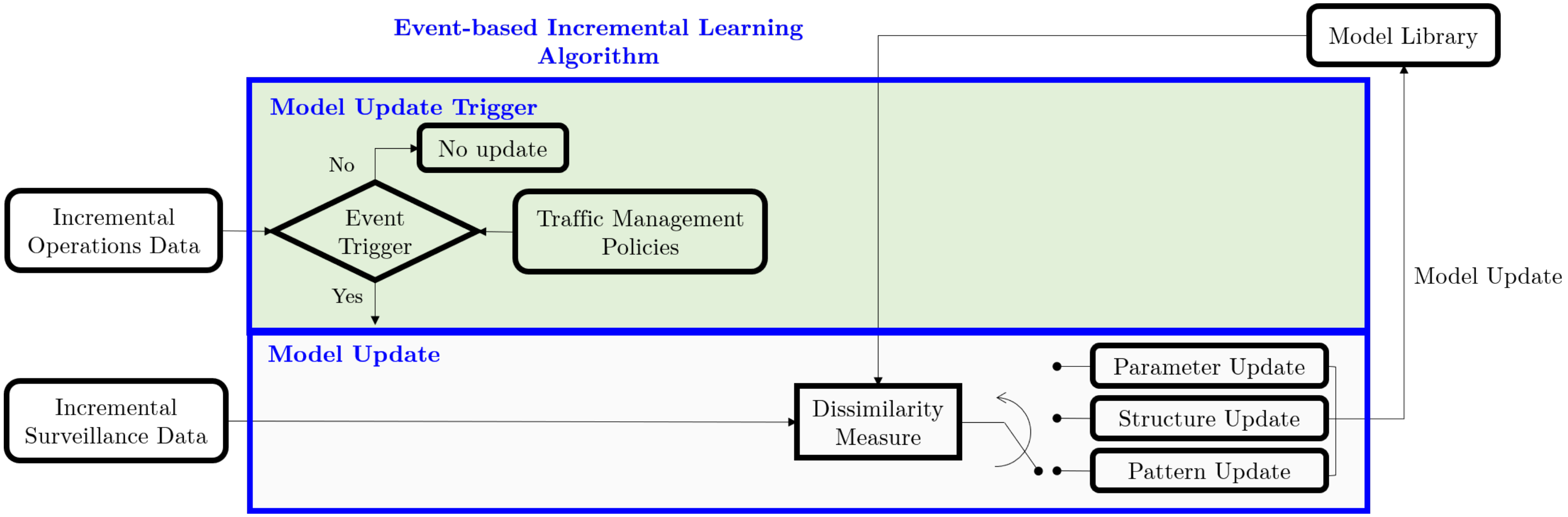
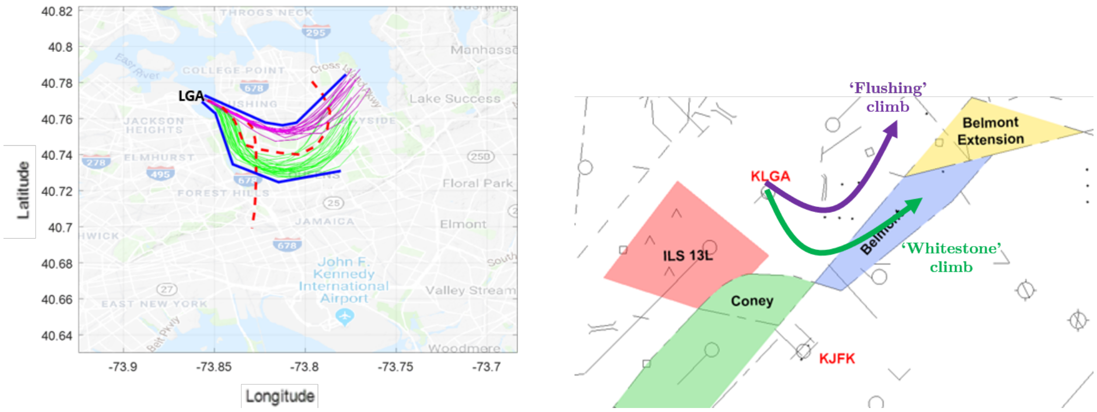


Figure 2.6. Update mechanism for event-triggered TempAD



(a)

(b)

Figure 2.7. 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).

Model update trigger: Operational events

As a motivating example, consider the departures from runway 13 at LGA during a day, as shown in Fig. 2.7(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, more accurate models can be generated, as shown in Fig. 2.8. Initially from 7:00 am to 8:59 am, both 'Coney' and 'Belmont' airspaces (shown in Fig. 2.7(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, first, the model at LGA for 'Flushing' climb can be gen-

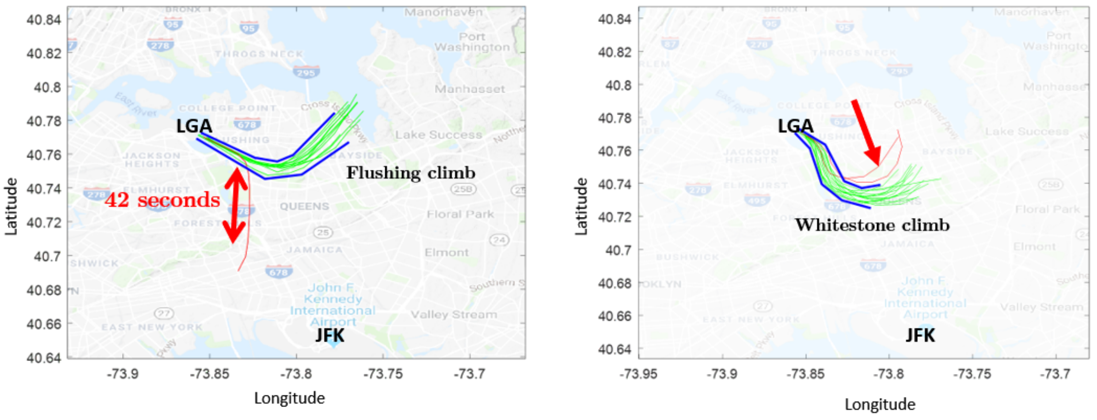


Figure 2.8. Horizontal anomaly detection models for all departures from LGA RWY13, segregated by climb paths (Left: ‘Flushing’ climb, right: ‘Whitestone’ climb)

erated, and when a change in JFK operating runways is observed, the model library for ‘Whitestone’ climb can then be updated, thus generating two distinct 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 seconds 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 2.1.1, the ASPM data recorded at the three airports within the New York metroplex is used to obtain the information about the operations at each airport. From the ASPM dataset, the following parameters are extracted 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, the FAA's strategic policies, or Standard Operating Procedures (SOPs) [48], are considered 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 [46].
- LGA and EWR: The combination of operating runways at JFK, EWR, and LGA, 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 [45].

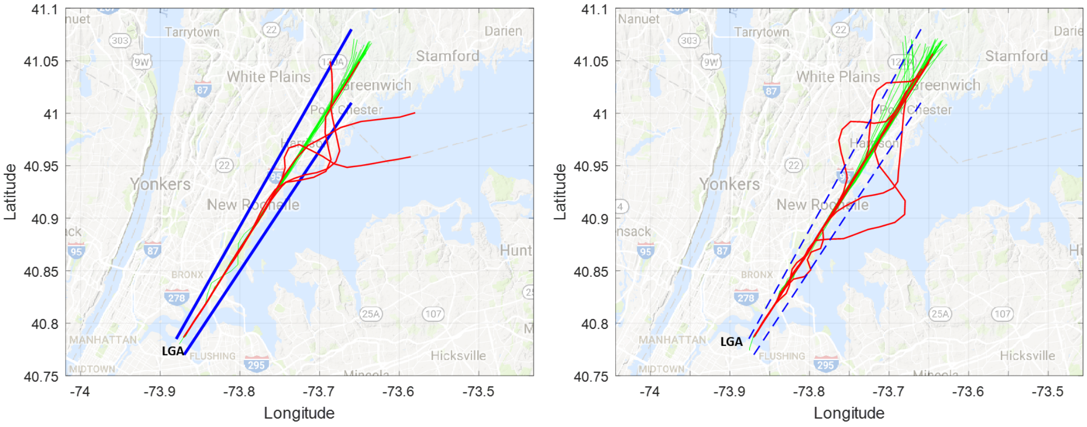
Each change in the operational parameters in the metroplex which can potentially affect the collective aircraft behaviors is an *operational event trigger*, after which the model library is required to be updated.

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. To develop such a metric – called *dissimilarity measure* – distinct observations are made for each update type.

Suppose that after the day when the initial model is generated, the incremental data has operations with a similar shape, as shown in Fig. 2.9. Only a few violations with low severity are observed, and thus, the initial model can describe the new data relatively well and needs only minor modification.

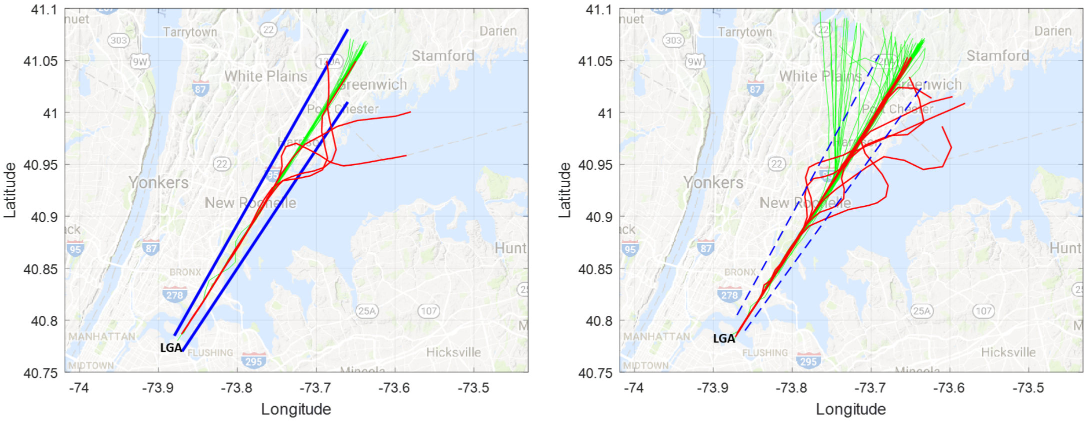
However, in Fig. 2.10, comparing the initial model and the incremental data, a large number of violations are observed, with a high severity of violations, which



Initial flight data and model

Overnight data and previous day model

Figure 2.9. Illustration showing requirement of a minor update of the anomaly detection model



Initial flight data and model

Overnight data and previous day model

Figure 2.10. Illustration showing requirement of a major update of the anomaly detection model

indicates that the initial model cannot account for the new data well and needs significant modification.

As mentioned earlier, the set of all observed models is stored in a model library, so that they can be selected during appropriate update. It is possible that on some days when arrival operations are varied significantly, none of the existing models can effectively describe the incremental data, and thus all the existing models identify nearly all new data as anomalous. Thus, a new model needs to be generated and added to this model set to accurately describe this incremental recording. This step is identical to the initial model design, as a syntax search and continuous parameter search needs to be performed from scratch, using the base-learning technique.

From the illustrations, it can be observed that each update type has different characteristics which can be expressed in terms of the number of violations that the incremental data $\{s_i\}_{i=1}^{N}$ has with respect to the old model φ_θ, and the degree of any such violations. Recall that for a given model φ_θ, the cost function $f\left(\{s_i\}_{i=1}^{N}, \varphi_\theta\right)$ in Eq. (2.2) represents (i) how many s_i's violate the model φ_θ by the tightness function $d\left(\{s_i\}_{i=1}^{N}, \varphi_\theta\right)$, and (ii) how severe each violation is by the slack variable $\mu\left(s_i, \varphi_\theta\right)$. Suppose there exists a model φ_θ learned from a past dataset $\{s_i\}_{i=1}^{N}$ and a newly recorded incremental dataset $\{s_i\}_{i=N+1}^{N+N'}$ (N' is the number of flights in the incremental dataset) is obtained. Let $P \coloneqq f\left(\{s_i\}_{i=1}^{N}, \varphi_\theta\right)$ and $Q \coloneqq f\left(\{s_i\}_{i=N+1}^{N+N'}, \varphi_\theta\right)$ 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 2.2.1). In this regard, the update type is determined according to the value of the dissimilarity measure, as follows:

$$\text{Update type} = \begin{cases} \text{Parameter update} & \text{if } \left|\frac{P-Q}{P}\right| < threshold_1 \\ \text{Structure update} & \text{if } threshold_1 < \left|\frac{P-Q}{P}\right| < threshold_2 \\ \text{New pattern update} & \text{if } \left|\frac{P-Q}{P}\right| > threshold_2 \end{cases}$$

- A small value of the dissimilarity measure implies that the existing model represents the incremental data well, and thus only an update in parameters (*parameter update*) is triggered.

- A large value for the dissimilarity measure implies 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*).

- An extremely large value for the dissimilarity measure implies 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. The update types are implemented in the following manner:

- Parameter update: If only a parameter update is required, the Adaptive One-class Support Vector Machine (AOSVM) algorithm [49] is used to determine the new continuous parameters based on the old model and new data.

- Structure update: If a structure update is required, new breakpoints for lower and upper bounds along with the number of piecewise linear predicates or sections that represent lower and upper bounds are found, and then the coefficients of each section are determined.

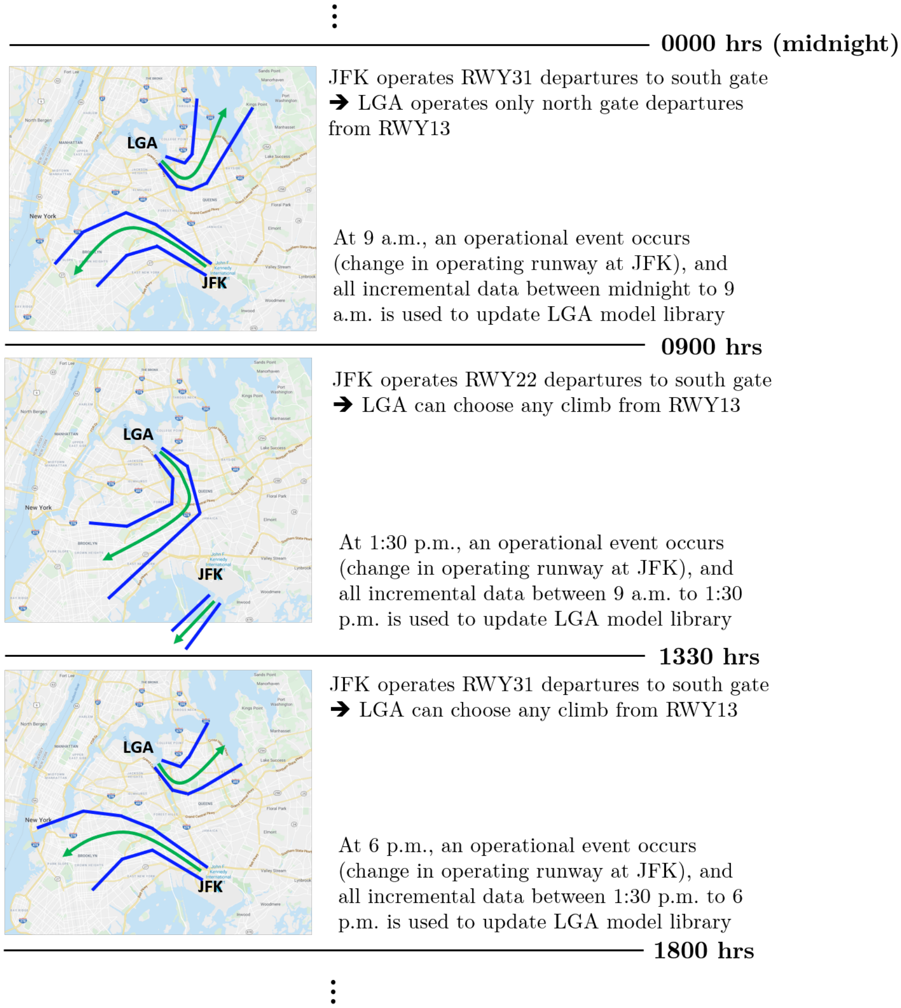
- New pattern update: If a new pattern update is required, a new anomaly detection model is developed from scratch.

This completes the design and development of the proposed algorithm for recursively generating anomaly detection models, whose update step summarized in Algorithm 1 and illustrated in Fig. 2.6.

Algorithm 1: Event-triggered temporal logic based anomaly detection algorithm

```
Result: Updated anomaly detection models
Input: Incrementally recorded data, model library;
Cluster all input data using DBSCAN;
while cluster with no model exists do
    Compute dissimilarity measure;
    if small variation in measure then
        Determine updated continuous parameters using adaptive one-class
          SVM;
    else if intermediate variation in measure then
        Determine new breakpoints and number of predicates using regression;
        Determine continuous parameters for each predicate;
    else
        Generate new model from scratch using syntax search and continuous
          parameter search, as in [50];
    end
end
```

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. 2.11.



⋮

0000 hrs (midnight)

JFK operates RWY31 departures to south gate
➔ LGA operates only north gate departures from RWY13

At 9 a.m., an operational event occurs (change in operating runway at JFK), and all incremental data between midnight to 9 a.m. is used to update LGA model library

0900 hrs

JFK operates RWY22 departures to south gate
➔ LGA can choose any climb from RWY13

At 1:30 p.m., an operational event occurs (change in operating runway at JFK), and all incremental data between 9 a.m. to 1:30 p.m. is used to update LGA model library

1330 hrs

JFK operates RWY31 departures to south gate
➔ LGA can choose any climb from RWY13

At 6 p.m., an operational event occurs (change in operating runway at JFK), and all incremental data between 1:30 p.m. to 6 p.m. is used to update LGA model library

1800 hrs

⋮

Figure 2.11. Event triggers for updating LGA model library

2.3 Test and analysis of event-triggered TempAD algorithm

In this section, first, a summary of results from extensive tests performed using the proposed event-triggered TempAD algorithm is presented. 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.

2.3.1 Summarized test results

Distribution of anomalies 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 2.2 and Table 2.3, 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 2.2 and 2.3:

- 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. 2.2.

- 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.

Table 2.2. 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 2.3. 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

- The rate of anomalies per flight is highest in the horizontal dimension. This can be attributed to the slightly relaxed constraints from ATC regarding these maneuvers, particularly as compared to the strict policies for approach altitude and speed.

Figure 2.12 shows the average rate of anomalies detected through the day plotted according to the hour they were detected in, while Fig. 2.13 shows the average distribution of arrivals through the day at each airport. Note that the average number of arrivals at LGA between midnight and 5:00 am is extremely small, but non-zero.

Figure 2.12 shows that for all three airports, the number of anomalies is higher in the afternoon, when the volume of arrival traffic is high in the metroplex area. Also, the number of anomalies does not scale linearly with the number of arrivals, i.e., a higher arrival traffic induces a much higher number of detected anomalies.

An interesting implication of the trends in Figs. 2.12 and 2.13 is related to how anomaly rates are affected by metroplex operations (i.e., coordination between nearby airports). Although LGA has a nearly constant arrival rate throughout the day between 7:00 am to 9:00 pm, as shown in Fig. 2.13, the anomaly rate spikes significantly after noon. This spike is correlated strongly with the increase in the number of arrivals at JFK and is observed on a day-to-day basis. Further analysis of results shows that there is a strong coordination strategy for airspace usage between LGA and JFK (note that LGA and JFK are separated only by 7.3 nautical miles), which is mandated in policies by FAA. Thus, this metroplex coordination strategy is an emergent behavior visible from the results of anomaly statistics at the airport.

Distribution of update types During anomaly detection, distribution of update types is also recorded. The strictly-controlled nature of arrivals in the horizontal domain makes it necessary to study how the event-triggered TempAD algorithm generates models and adapts to the variations in arrival operations as days progress. Table 2.4 presents this distribution of update types across the 3 months recording obtained from the TAIS dataset.

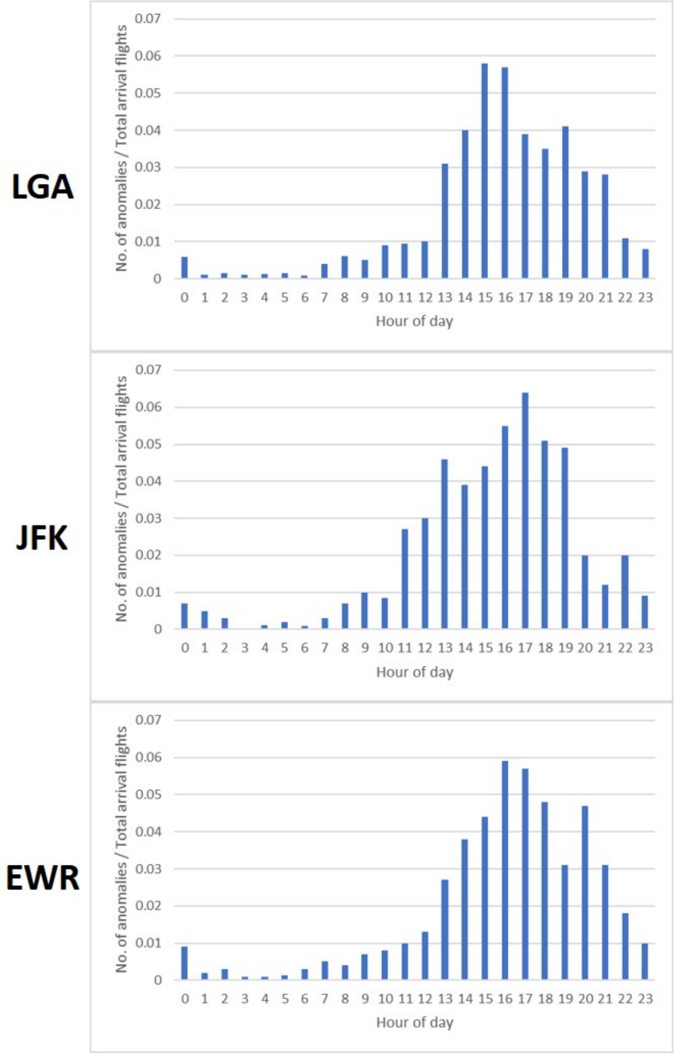


Figure 2.12. Distribution of anomaly rate by time of day

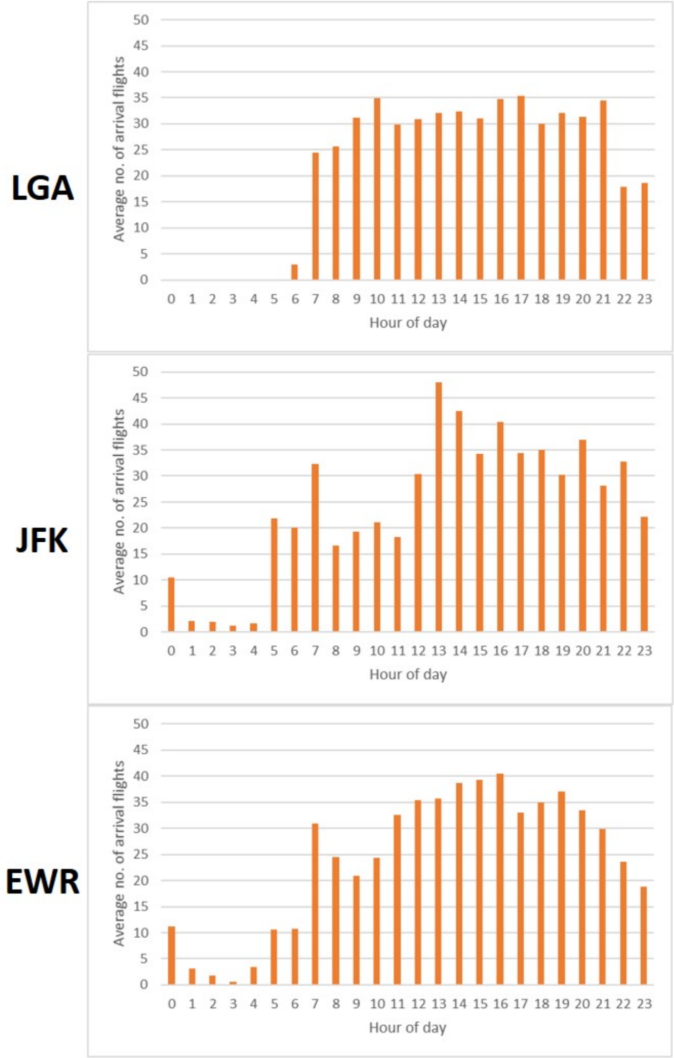


Figure 2.13. Average number of arrival flights by time of day

Table 2.4. Distribution of update types for each dimension, within arrivals

Dim.	**Airport**	**No. of new pattern upd.**	**No. of param. upd.**	**No. of struct. upd.**	**No. of distinct structs.**
H	LGA	10	384	71	17
	JFK	10	479	75	15
	EWR	13	428	69	18
V	LGA	10	414	36	14
	JFK	10	492	40	17
	EWR	13	423	34	15
S	LGA	10	389	85	16
	JFK	10	512	84	16
	EWR	13	437	88	19
STE	LGA	10	512	37	16
	JFK	10	484	53	18
	EWR	13	425	39	15
SPER	LGA	10	399	47	12
	JFK	10	436	51	16
	EWR	13	481	38	17

param. → parameter
struct. → structure
upd. → updates

A critical insight regarding the structure updates is that they imply a significant change in procedures adopted by the ATC during the day. There are fewer structure updates in the vertical dimension as compared to the horizontal and speed dimensions. This could be attributed to ATC guidelines being more static and rigid for the vertical approach as well as having fewer candidate waypoints in the vertical dimension, for arrivals.

During the initial few days, while running the anomaly detection algorithm, numerous new pattern updates are triggered, as the model library ingests new approaches and becomes richer. After a few days, only structure updates and parameter updates are observed. Among these, parameter updates are more frequent, and drastic changes in the operational structure are rare. The frequency of update types as the algorithm runtime (amount of recorded data ingested) increases is presented in Table 2.5, which shows the number of model updates of each type observed in the horizontal dimensions during those specific dates, for LGA arrivals.

Table 2.5. Frequency of update types, within arrivals

Date range	**Param. updates**	**Struct. updates**	**New pattern updates**
Sept. 1 – 9	36	4	3
Sept. 10 – 19	42	2	6
Sept. 20 – 29	59	13	0
Sept. 30 – Oct. 9	34	7	0
Oct. 10 – 19	37	5	1
Oct. 20 – 29	23	11	0
Oct. 30 – Nov. 9	35	12	0
Nov. 10 – 19	48	10	0
Nov. 20 – 28	17	7	0

param. → parameter
struct. → structure

As expected, parameter and structure updates are present throughout the entire recording period, but most of the new pattern updates (when a new approach is observed for the first time) are during the earlier days of training.

The last column in Table 2.4 shows the number of distinct shapes that the discrete structures of the anomaly detection models take throughout the data recording period.

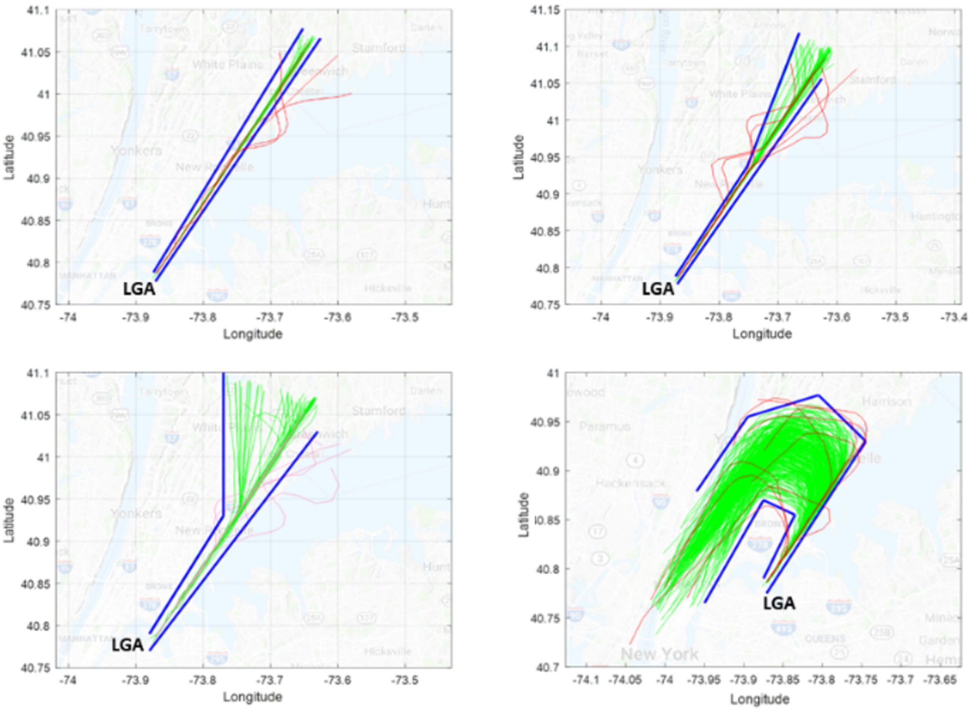


Figure 2.14. Discrete shapes for LGA approach to RWY22

These discrete shapes visually show the navigation corridors that the ATC uses to maneuver aircraft during the short final.

Illustrations of a few such discrete shapes that identify the navigation corridors in the horizontal dimension for LGA arrivals are presented in Figs. 2.14 and 2.15. For example, for flights landing at runway 22 at LGA, the coefficients of the predicates may change daily, but the general shape of the entire model switches between the illustrated shapes.

Types of anomalies

All anomalies detected in the basic dimensions can be summarized and characterized into a small number of representative anomalies that occur across all three airports. Such representative illustrations of anomalies that are common across all

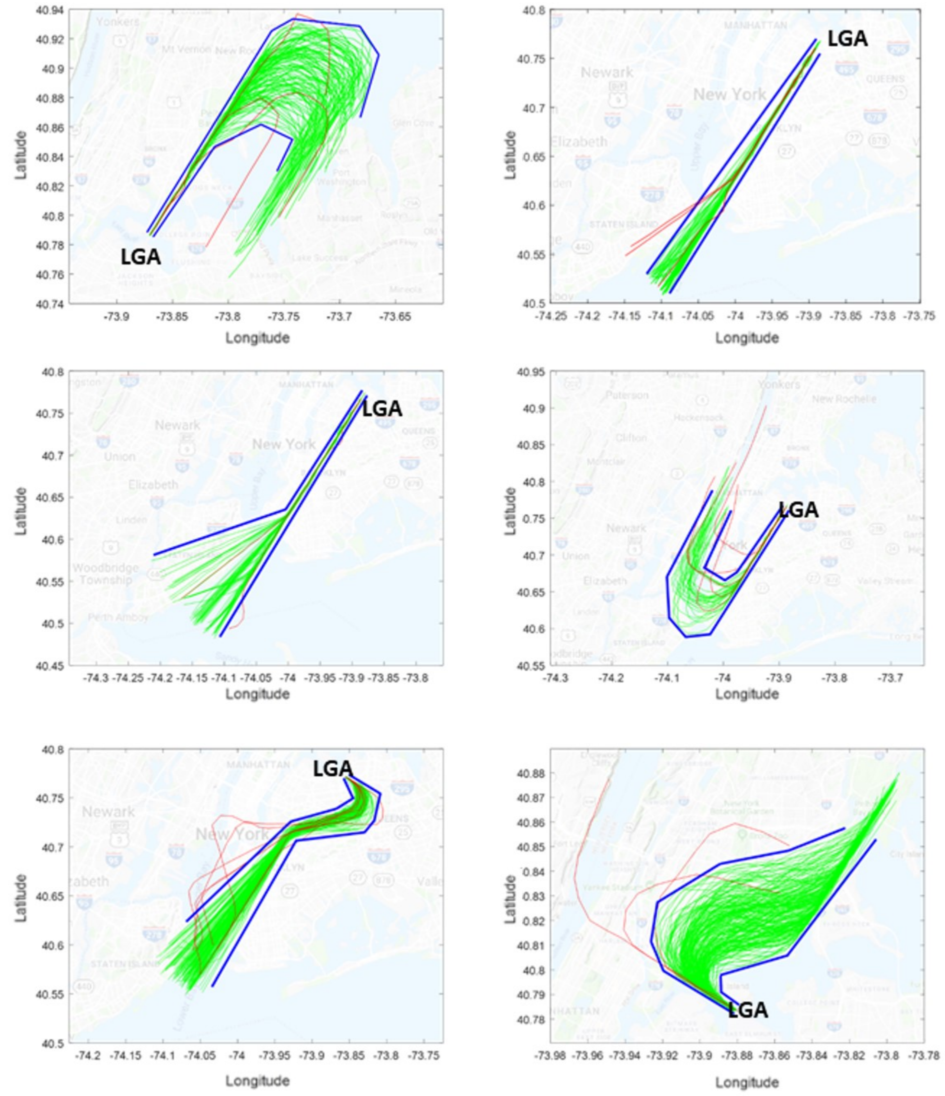


Figure 2.15. Discrete shapes for LGA approach

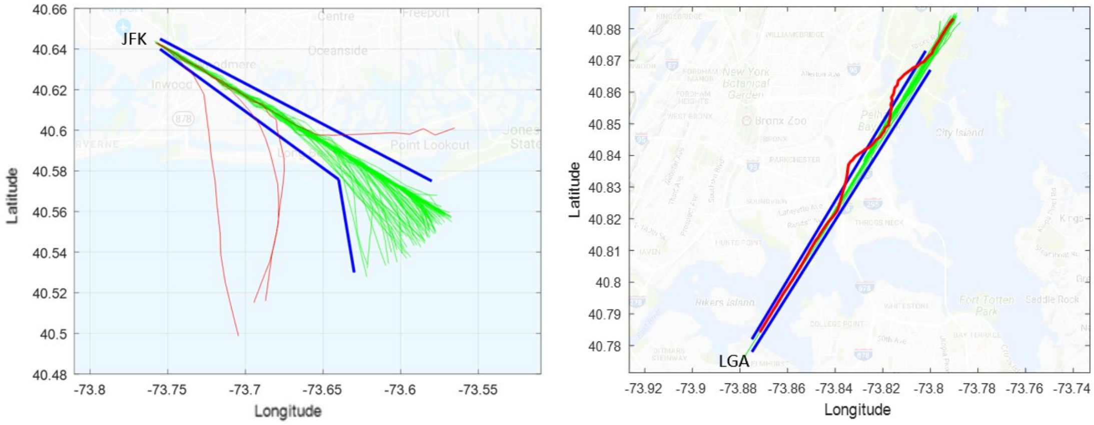


Figure 2.16. Horizontal anomaly detection: late transition from en-route airspace to short final (left) and repeated horizontal deviations (right)

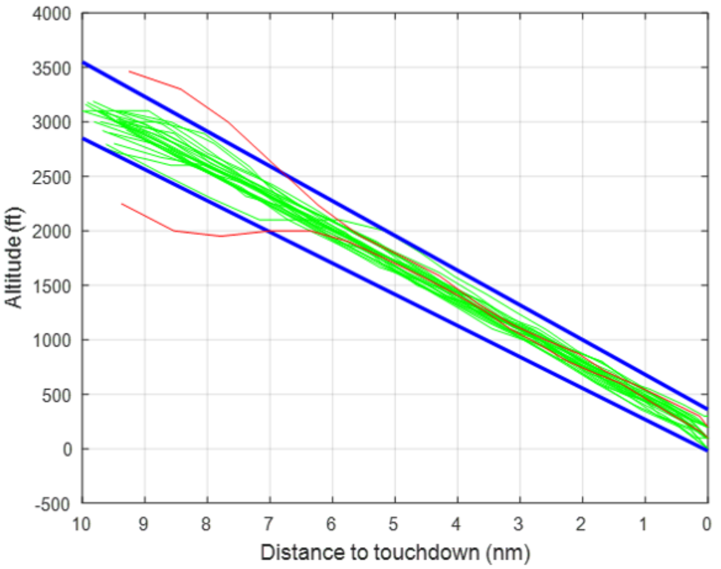


Figure 2.17. Vertical anomaly detection: Late capture of glide slope

three airports are presented. Figures 2.16, 2.17 and 2.18 present anomaly detection models in the horizontal, vertical and speed dimensions, respectively.

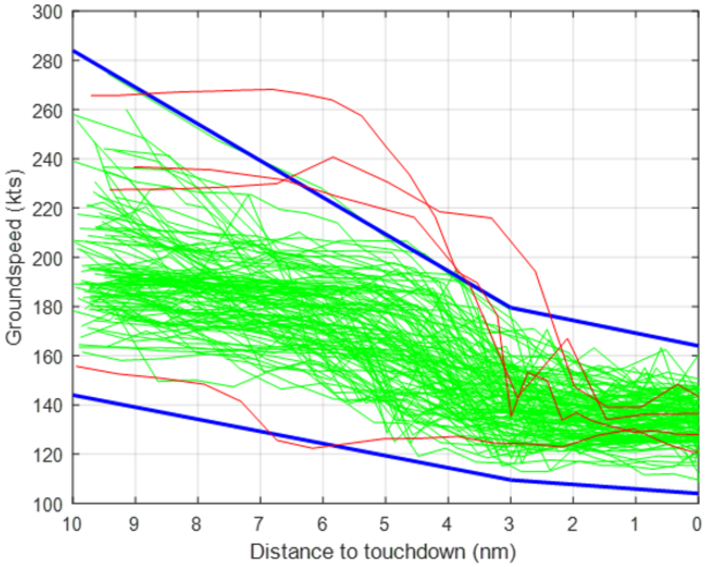


Figure 2.18. Speed anomaly detection: Excessive or deficient speed

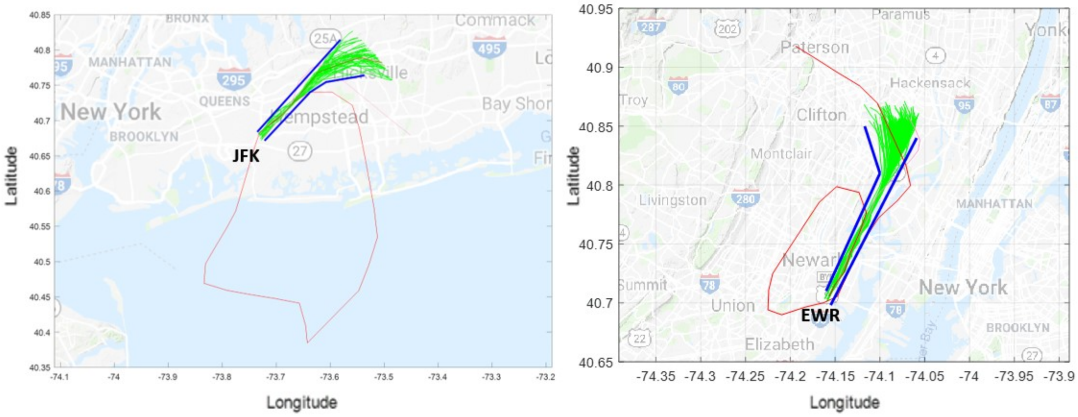


Figure 2.19. Detection of go-around anomalies at JFK (left) and EWR (right)

In Fig. 2.16, the left subfigure shows 4 anomalous flights that delayed the transition from en-route airspace to the short final. These flights merge with the stream of arriving flights much later as compared to other flights. On the other hand, the right subfigure presents an anomalous flight which undertakes an S-turn maneuver (repeated horizontal deviations). Such a maneuver is generally undertaken to reduce the distance from the preceding flights or to reduce airspeed without losing significant altitude [51]. Figure 2.17 illustrates two anomalous flights in the vertical dimensions, where the anomalous flights intercept the glide slope at higher-than-normal or lower-than-normal angles. Figure 2.18 presents anomalous flights that have an excessive or deficient speed during the short final. This could be an operational issue, as it could result in higher landing distance and extra stress on the airframe during landing, and also result in the flight occupying the landing runway for a higher duration.

Figure 2.19 illustrates the detection of a missed approach/go-around anomaly by the horizontal anomaly detection model at JFK and EWR airports.

Figure 2.20 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. Figure 2.21 shows the STE anomaly

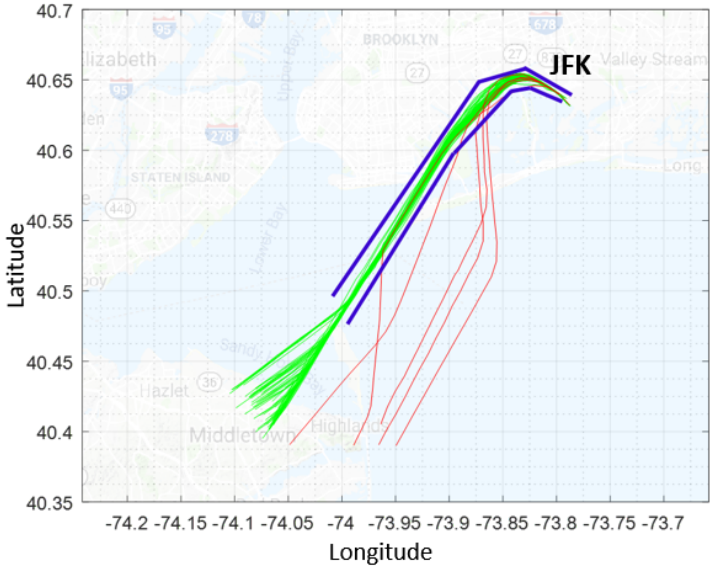


Figure 2.20. Horizontal anomaly detection model for departures from JFK RWY31R

detection model for departing flights, where the flight with excess STE is detected as an anomaly.

2.3.2 Study of the impact of operations on 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, and the impact of the operations data is emphasised.

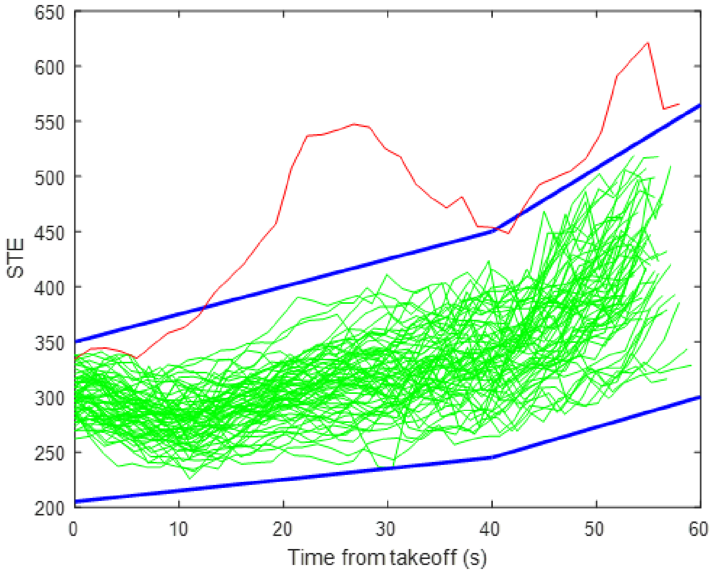


Figure 2.21. Specific total energy anomaly detection model for departures from LGA RWY22

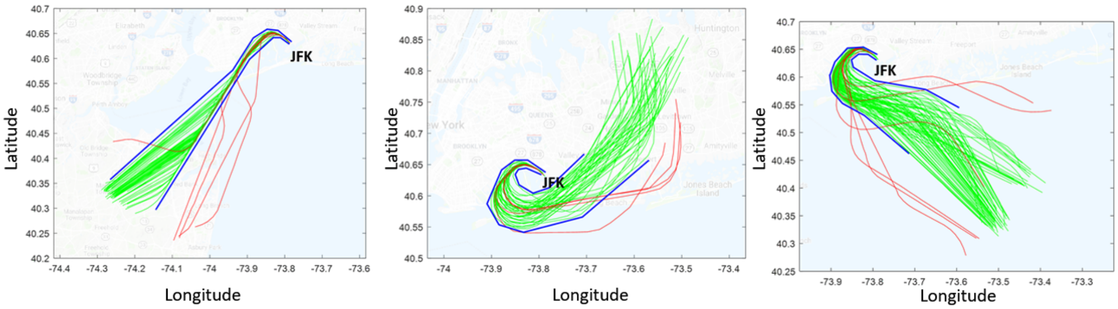


Figure 2.22. Horizontal anomaly detection models at JFK, generated using event-triggered TempAD (southwest gate, east gate, and water gate, respectively)

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 Upon using event-triggered TempAD to detect anomalies (in red), the framework presented in Fig. 2.6 relies on the departure exit gate feature to generate more effective anomaly detection models, as shown in Fig. 2.22. The departure exit gates mandated by FAA [46] are presented in Fig. 2.23. The anomalies here correspond to the trajectories that do not follow a designated climb path.

Vertical anomaly detection Figure 2.24 presents the vertical anomaly detection models generated for departures using event-triggered TempAD. The effectiveness of the anomaly detection models generated by the proposed algorithm is demonstrated by how well the anomaly detection models mimic the policies mentioned in the FAA documents. For the east gate departures, operational policies mandate a minimum altitude of 2,500 feet at 6 nautical miles from takeoff. The event-triggered TempAD

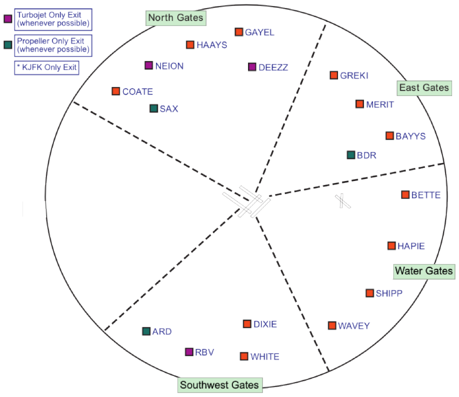


Figure 2.23. JFK departure gates

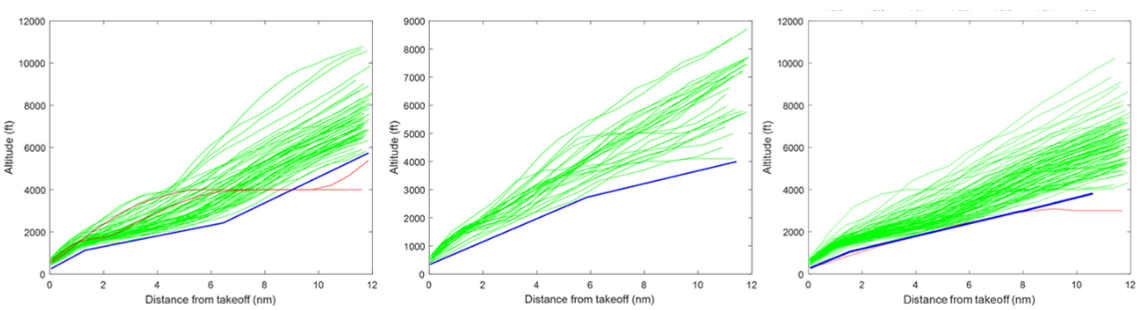


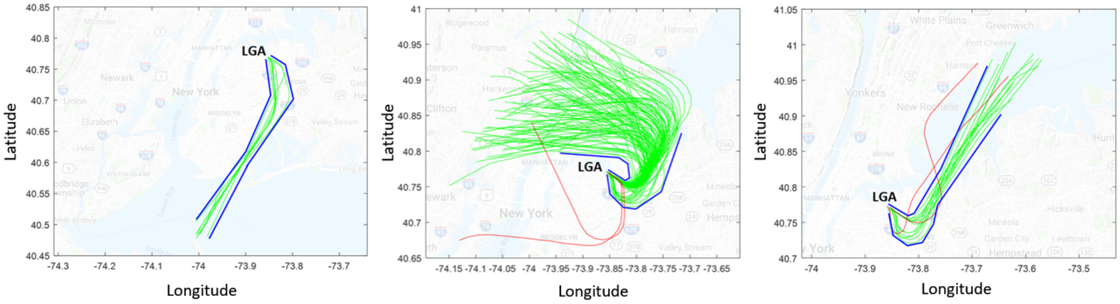
Figure 2.24. Vertical anomaly detection models at JFK, generated using event-triggered TempAD (southwest gate, east gate, and water gate, respectively)

model shows a lower bound of 2,700 feet, and thus, the safety bound given by operational policies is successfully captured using the event-triggered TempAD algorithm.

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 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. 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 [45], it is observed that the departure models segregate into the south, west, and north gate departures, as shown in Fig. 2.25, and are capable of detecting horizontal operational anomalies effectively.

A similar analysis can be performed for EWR operations as well. Figure 2.26 illustrates the horizontal anomaly detection models generated for all departures from

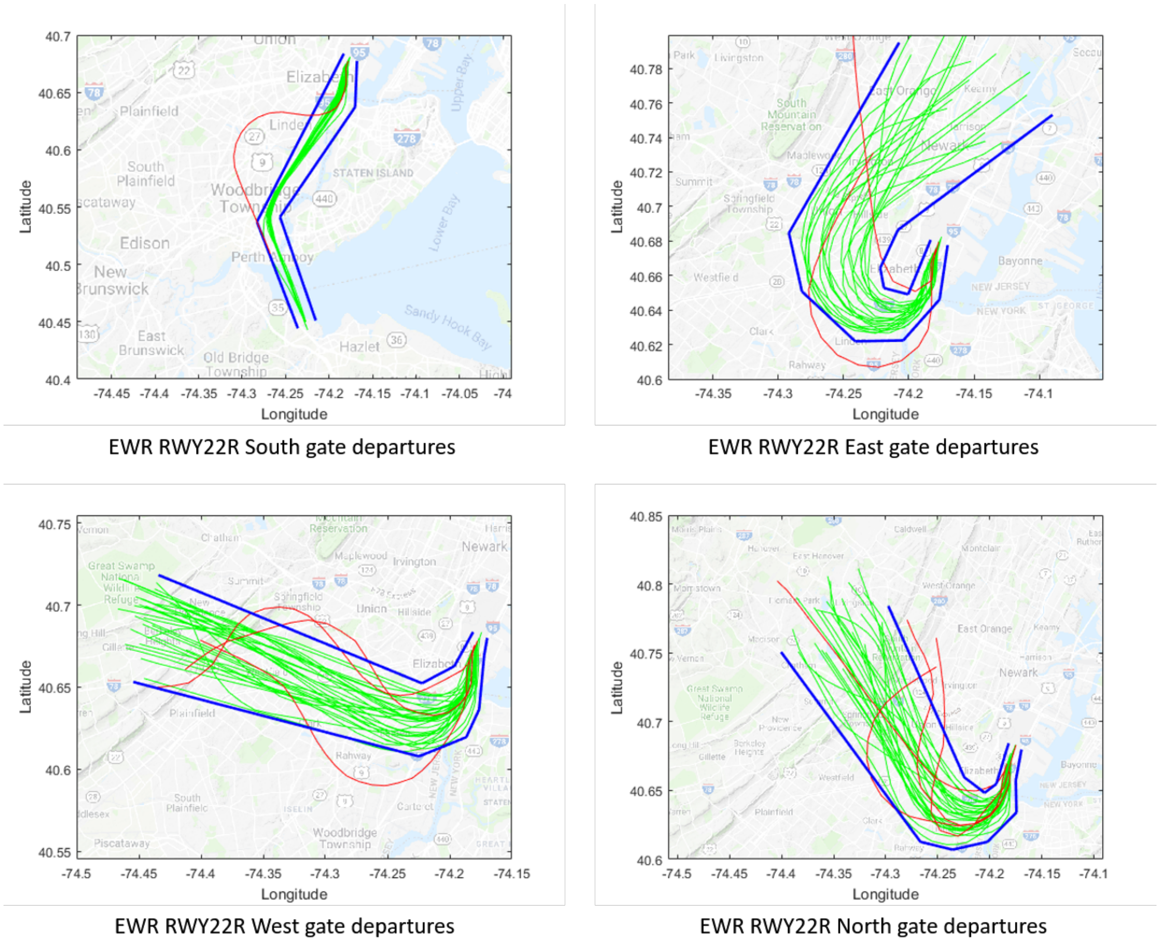


South gate departures
JFK operating 22L/R
(12:00 a.m. → 11:30 a.m.)

West gate departures
JFK operating 4L/R
(11:30 a.m. → 2:00 p.m.)

North gate departures
JFK operating 22L/R
(2:00 p.m. → 12:00 a.m.)

Figure 2.25. Horizontal anomaly detection models at LGA, generated using event-triggered TempAD (south gate, west gate, and north gate, respectively)



EWR RWY22R South gate departures

EWR RWY22R East gate departures

EWR RWY22R West gate departures

EWR RWY22R North gate departures

Figure 2.26. Horizontal anomaly detection models for departures from EWR RWY22R, generated using event-triggered TempAD

runway 22R at EWR during an entire day, wherein multiple models are generated using the departure gates feature in the event-triggered TempAD algorithm.

2.3.3 Degree of anomalousness and extreme anomalies

All anomalies detected by the event-triggered TempAD algorithm are outliers and depart from normal behavior, and thus could be either safety threats or operational anomalies. However, not all detected anomalies are actual safety threats: some of them significantly violate the model generated, but others only slightly deviate from the model. The design of the algorithm includes the robustness degree (r), which is a signed distance measure of the data signal s_i from the predicate φ_θ. r has a more negative value if a flight is further away from predicate and violates the model more severely. By normalizing r between 0 (least anomalous) and 1 (most anomalous), the degree of anomalousness $\hat{r}$ can be computed. This $\hat{r}$ describes how significantly the flight violates the model, thereby telling which anomalous flight could be more operationally significant than others. Anomalies that have statistically extreme values for $\hat{r}$ within each dimension are specifically studied and analyzed when examining data from all days in the test set.

For example, Fig. 2.27 shows the horizontal anomaly detection model for arrivals at runway 22 at LGA (in blue). This model has detected four anomalies, labeled 1 through 4. For each of these anomalies, the degree of anomalousness is:

- Anomaly 1: $\hat{r} = 0.31$
- Anomaly 2: $\hat{r} = 0.91$
- Anomaly 3: $\hat{r} = 0.53$
- Anomaly 4: $\hat{r} = 0.47$

Among these 4 anomalies, Anomaly 2 has a very high degree of anomalousness, and is thus tagged as an extreme anomaly. Similar detections of extreme anomalies in other basic dimensions are presented in Fig. 2.28.

The search for extreme anomalies is important as these may be the riskiest occurrences. However, not all extreme anomalies are operationally dangerous, as they may be correctable and the aircraft may be stabilized if they occur far from an airport, but some extreme anomalies that occur close to the ground are dangerous and can result in an accident, and are thus operationally significant. Potentially hazardous anomalies are detected by searching for anomalies with a high degree of anomalousness during the final 3 nautical miles before touchdown. This includes trajectories that violate the lower bound of the vertical model (prone to flight into terrain) as

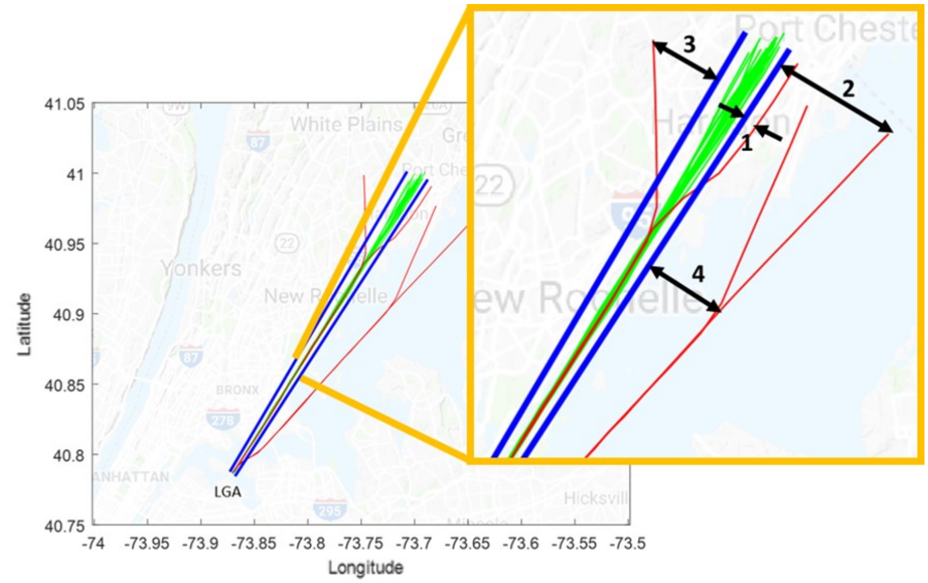


Figure 2.27. Horizontal anomalies for LGA RWY22 arrivals

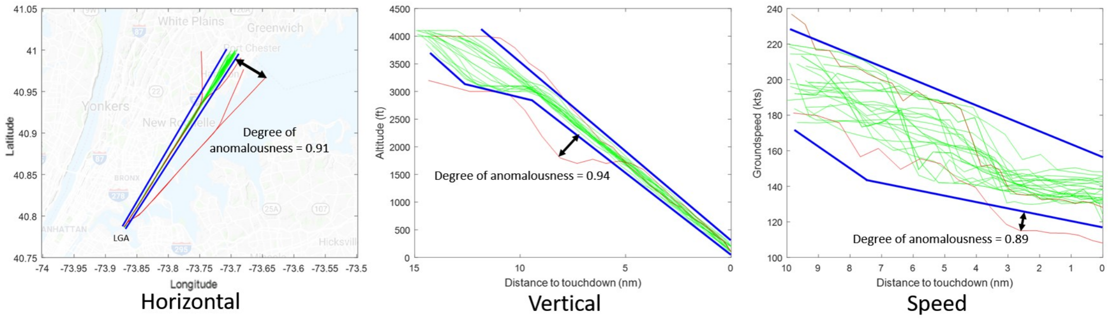


Figure 2.28. Extreme anomalies in horizontal, vertical and speed dimensions for LGA RWY22 arrivals

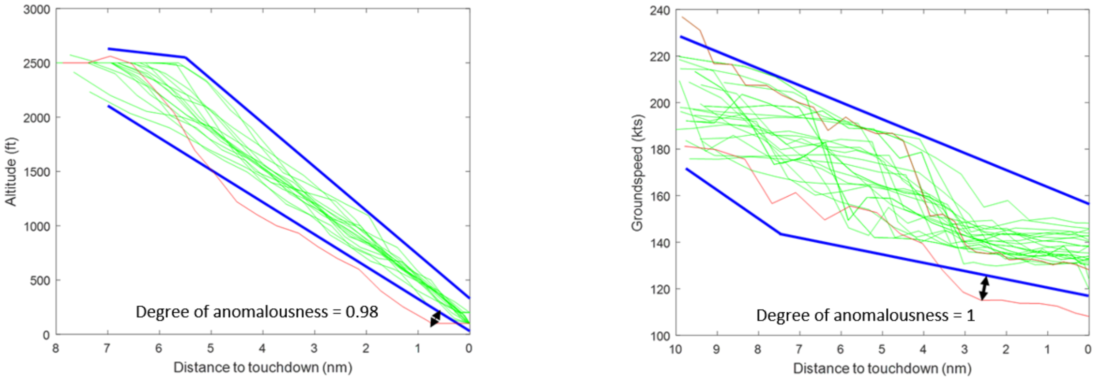


Figure 2.29. Operationally significant anomalies

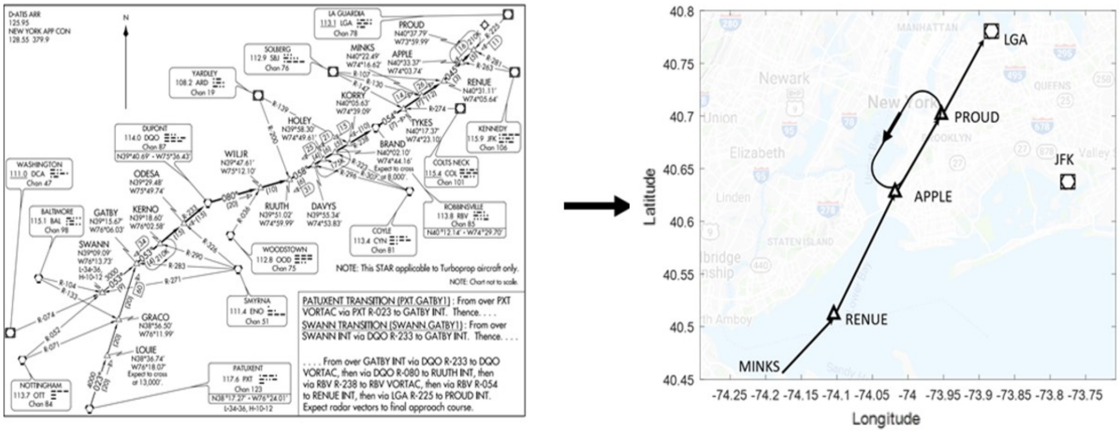
shown in Fig. 2.29 (left), or flights that violate the lower bound of the speed anomaly detection model (prone to stalling) as shown in Fig. 2.29 (right). These anomalies are thus recorded as potential operationally dangerous anomalies.

Therefore, using the degree of anomalousness as an anomaly score shows the usefulness of the proposed algorithm for identifying operationally significant anomalies and obtaining human feedback.

2.3.4 Identification of airspace structure

It was observed from horizontal anomaly detection models generated from the event-triggered TempAD algorithm can infer the airspace structure (navigation waypoints) used by the ATC to manage and route aircraft to the airport. For arrivals, these waypoints are recorded in the STAR (Standard Terminal Arrival Route) aeronautical charts. To illustrate this, the STAR chart for LGA [45] is used, and extract useful information specific to trajectories and discrete shapes observed in the available data, as shown in Fig. 2.30.

Comparing this chart and the anomaly detection models computed by the proposed algorithm for this approach, a close overlap is observed between the navigational



NOTE: This STAR applicable to Turboprop aircraft only.

NOTE: Chart not to scale.

PATUXENT TRANSITION (PXT.GATBY1) : From over PXT VORTAC via PXT R-023 to GATBY INT. Thence. . . .

SWANN TRANSITION (SWANN.GATBY1) : From over SWANN INT via DQO R-233 to GATBY INT. Thence. . . .

. . . . From over GATBY INT via DQO R-233 to DQO VORTAC, then via DQO R-080 to RUUTH INT, then via RBV R-238 to RBV VORTAC, then via RBV R-054 to RENUE INT, then via LGA R-225 to PROUD INT. Expect radar vectors to final approach course.

Figure 2.30. STAR chart for LGA for approaches from southwest direction

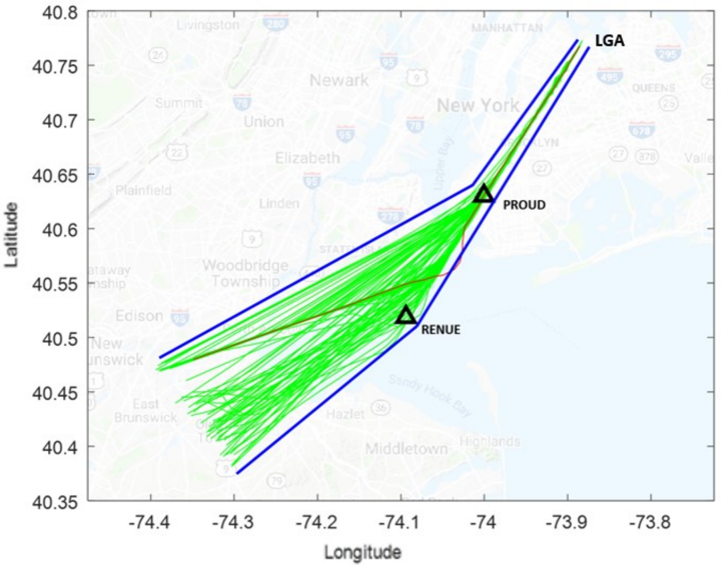


Figure 2.31. Horizontal anomaly detection models for LGA RWY4 approach: Model breakpoints overlap with RENUE and PROUD waypoints

waypoints and the model breakpoints. Illustrative examples of this close overlap are presented in Fig. 2.31 and Fig. 2.32.

Together, the anomaly detection models in these figures can correctly capture the STAR navigation waypoints within the last 20 nautical miles for arrivals to LGA runway 4. This observation extended to approaches from most directions for all three airports.

Similarly, in the vertical dimension, it was observed that the shape of the anomaly detection models corresponded closely with the waypoints that are recorded in the VGSI (Vertical Glide Slope Indicator) aeronautical charts. For demonstration, an example of the overlap of the vertical waypoints [45, 46] with the breakpoints in the vertical anomaly detection model is presented in Fig. 2.33.

From this analysis, it can be seen that the proposed algorithm is capable of learning the airspace operations and route structures from the surveillance data alone, without

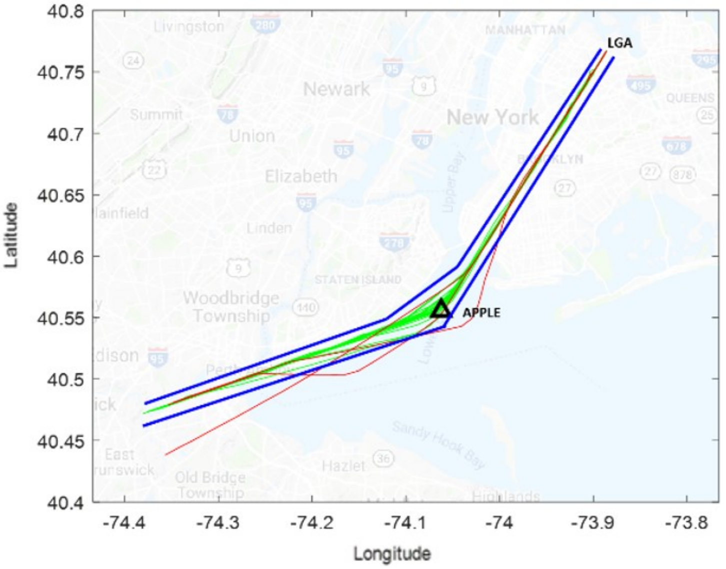


Figure 2.32. Horizontal anomaly detection models for LGA RWY4 approach: Model breakpoint overlaps with APPLE waypoint

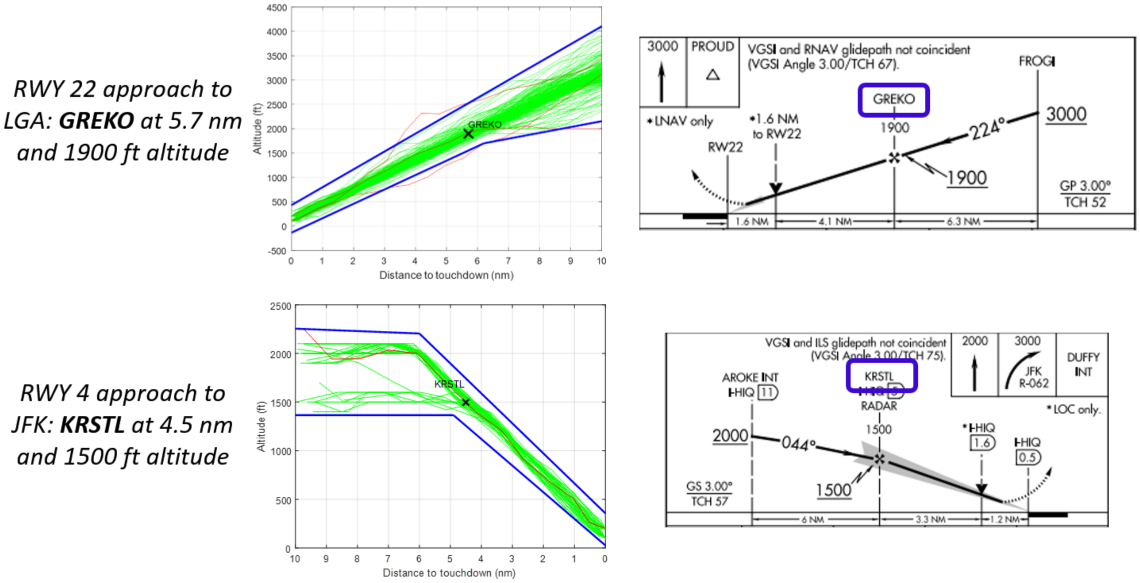


Figure 2.33. Vertical waypoints GREKO and KRSTL, with arrivals at LGA RWY22 and JFK RWY4 respectively, with vertical anomaly detection models

any prior map input. This proves that the proposed algorithm can be extended to any unknown airport to learn its operations through unsupervised learning.

2.4 Anomaly monitoring

The event-triggered anomaly detection algorithm is designed to use any historical surveillance and operations dataset as input, and thus, the entire trajectory and states for every aircraft is available beforehand. For real-time applications of any machine learning algorithm, modifications need to be made to the algorithm to accept and process streaming data, as there is only partial information available up to the current time and no knowledge of the future evolution of the time-series data is available. For example, for streaming data from an arriving flight, the knowledge of the approach path or landing runway may not be available, but the appropriate model from the model library still needs to be selected for anomaly detection. To circumvent this lack of future knowledge of the flight, a model inference approach is augmented to the algorithm framework to determine the most likely model that the flight will follow, generated using statistical methods. After the most likely model is inferred, the robustness degree is used to determine if a flight is anomalous or normal. The proposed framework for online (real-time) anomaly monitoring algorithm (henceforth called the online implementation) is presented in Fig. 2.34.

2.4.1 Model inference

For real-time anomaly detection, the model inference block attempts to infer an appropriate model from the model library for every arriving or departing flight. Assuming that the input to the algorithm is terminal airspace surveillance information comprising of the position (latitude and longitude), speed, altitude, timestamps and the unique identifier for every flight (e.g., from the TAIS surveillance dataset), a measure of *belongingness* is needed to associate a flight to the models. This measure is

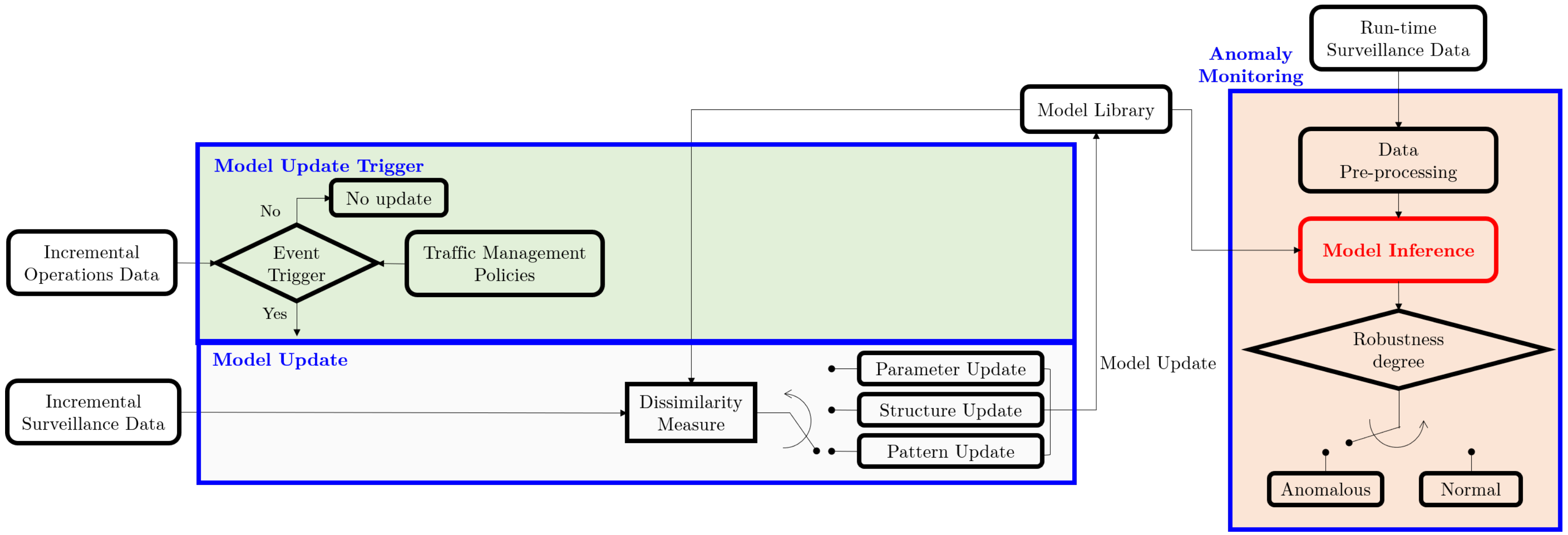


Figure 2.34. Framework of anomaly monitoring algorithm

used to develop a probabilistic solution, i.e., it determines the probability of the flight being associated with each model.

This measure of belongingness (called the model likelihood) is computed through the following steps:

- For each model, a Euclidean radius parameter is determined. This radius parameter is similar to the cluster radius used in the DBSCAN algorithm. Also, the centroid of the cluster is determined. Both are obtained in a time-series format.
- Now, the datapoint of the flight under consideration is obtained, as it is streamed in real-time and determine the radius and centroid of all clusters at the same radial distance as this flight.
- Then, the belongingness of the current datapoint to each cluster using likelihood is computed and then normalized to obtain probability.

Note that this probability is dynamic and changes as time progresses, i.e., some models become more likely as time progresses. To narrow down the search space for model candidates, the policies in the FAA documents [45–47] are used to correlate usable runways depending on the arrival gate.

During preliminary tests of the proposed model inference approach, for most cases, it was observed that model inference could be indeterminate if a flight is far from the airport, but then the true model (the true approach that the flight took) quickly becomes the most likely as the flight approaches to the airport. An example of the evolution of these probabilities is presented for a specific arrival to runway 4 at LGA using Figs. 2.35 and 2.36.

Figure 2.35 presents the individual centroids of the horizontal trajectories for the flights within the three candidate models. Initially, all three models are nearly equally likely, and the model inference block is confused over the choice of the model. However, as time progresses, the purple model (note that this is the true model) becomes the most likely model as shown in Fig. 2.36, since the distance of the flight

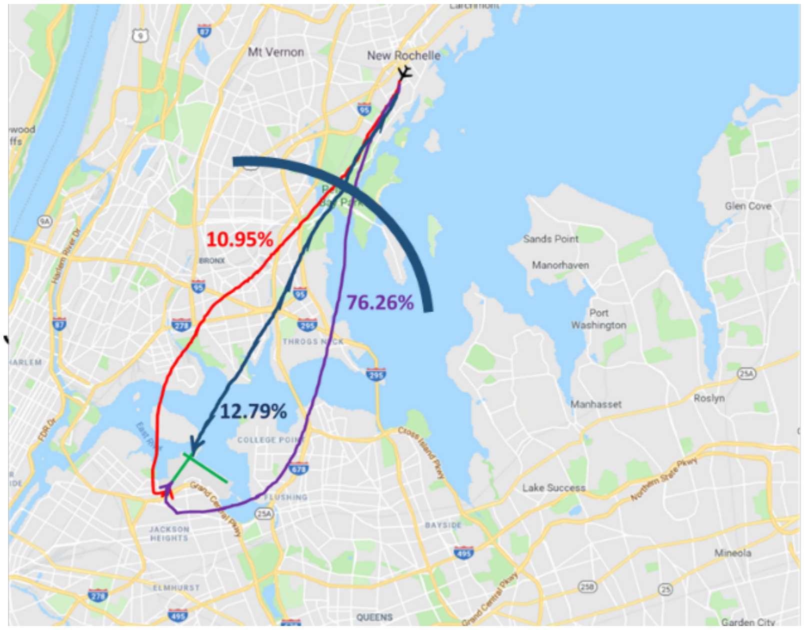


Figure 2.35. Candidate models for arrivals to LGA from the northeast

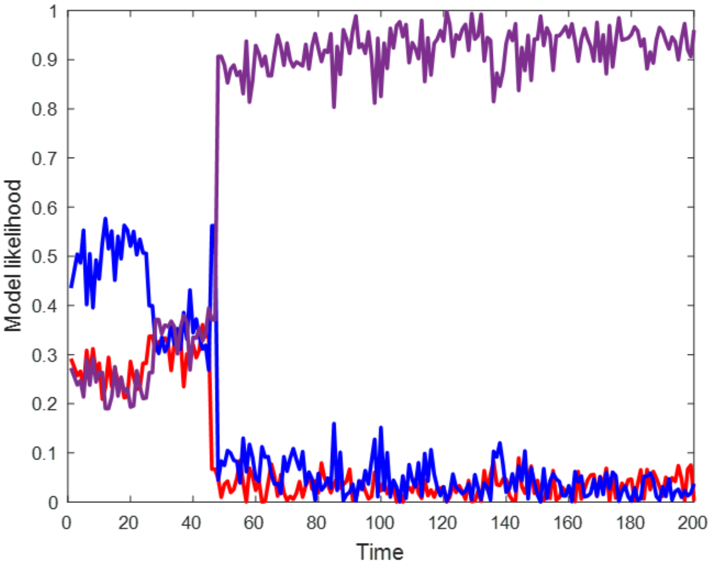


Figure 2.36. Evolution of model likelihoods over time

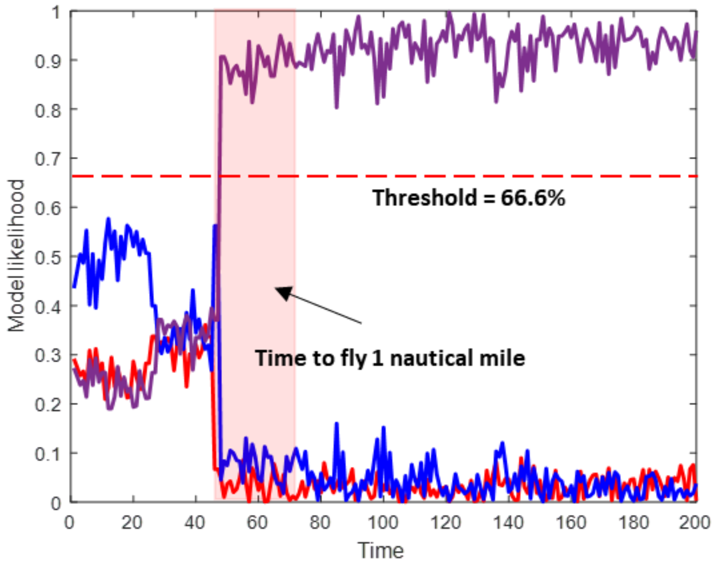
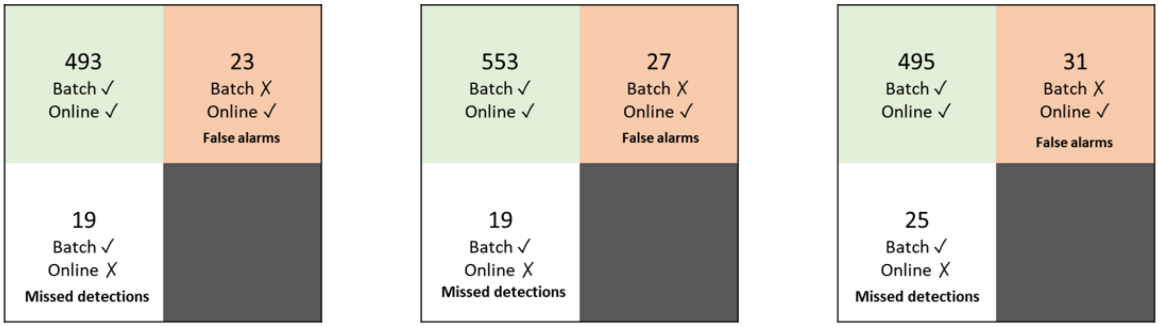


Figure 2.37. Thresholding for model inference

from the centroids of the blue and red models increases significantly (thereby rapidly decreasing their likelihoods of being true), which corresponds to the spike that occurs at time-step 48 (7.8 nautical miles away from the airport).

With extensive testing, it was observed that once a model becomes the most likely model with a high probability, it remains the most likely model until landing, or until reaching the departure fix. Thus, the computational load of the algorithm can be reduced by terminating the model inference block once there is assurance of the model the flight belongs to. A likelihood threshold is set, such that if any model crosses the threshold and continues to belong to that model for a flying distance of 1 nautical mile, that model is assigned to the flight permanently. The threshold is chosen as $\left(\frac{N-1}{N}\right)\%$ value of model likelihood, where N is the number of candidate models. This technique of circumventing the computational load is demonstrated in Fig. 2.37 for the flight presented in Fig. 2.35.



493 Batch ✓ Online ✓	23 Batch ✗ Online ✓ **False alarms**
19 Batch ✓ Online ✗ **Missed detections**	

553 Batch ✓ Online ✓	27 Batch ✗ Online ✓ **False alarms**
19 Batch ✓ Online ✗ **Missed detections**	

495 Batch ✓ Online ✓	31 Batch ✗ Online ✓ **False alarms**
25 Batch ✓ Online ✗ **Missed detections**	

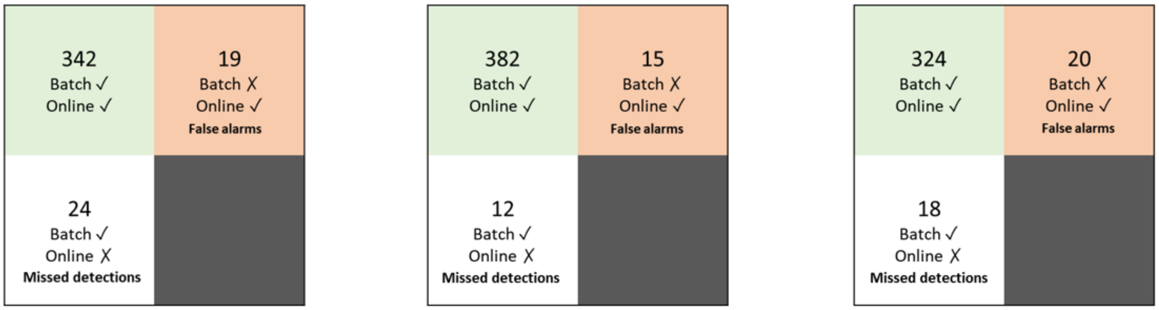
Figure 2.38. Confusion matrix of online anomaly monitoring for arrivals to LGA, JFK and EWR

Here, the number of candidate models is 3 (i.e., N=3). Thus, when the purple model achieves a likelihood of 66.6% for 1 nautical mile, the purple model is assigned to this flight and further model likelihood computations are stopped.

2.4.2 Test and analysis of anomaly monitoring algorithm

Extensive tests were performed using the framework for online anomaly detection inspired from event-triggered TempAD, for arrivals and departures within the New York metroplex recorded in the TAIS dataset between September to November 2016. To simulate real-time anomaly monitoring, the historical recorded data from November 2016 was streamed second-by-second as an input. September and October 2016 recordings were used as the training set. The results of this *online* test were compared against the *batch* case, where the entire November dataset is also considered as historical, and anomalies are detected using the earlier event-triggered TempAD algorithm. The confusion matrices for arrivals and departures for the three airports are presented in Figs. 2.38 and 2.39, respectively.

Considering that missed detections in a real-time setting are important and specifically critical in a safety-intensive application like air traffic management, anomalies that were missed by the online anomaly monitoring algorithm but were captured by



342 Batch ✓ Online ✓	19 Batch ✗ Online ✓ **False alarms**
24 Batch ✓ Online ✗ **Missed detections**	

382 Batch ✓ Online ✓	15 Batch ✗ Online ✓ **False alarms**
12 Batch ✓ Online ✗ **Missed detections**	

324 Batch ✓ Online ✓	20 Batch ✗ Online ✓ **False alarms**
18 Batch ✓ Online ✗ **Missed detections**	

Figure 2.39. Confusion matrix of online anomaly monitoring for departures from LGA, JFK and EWR

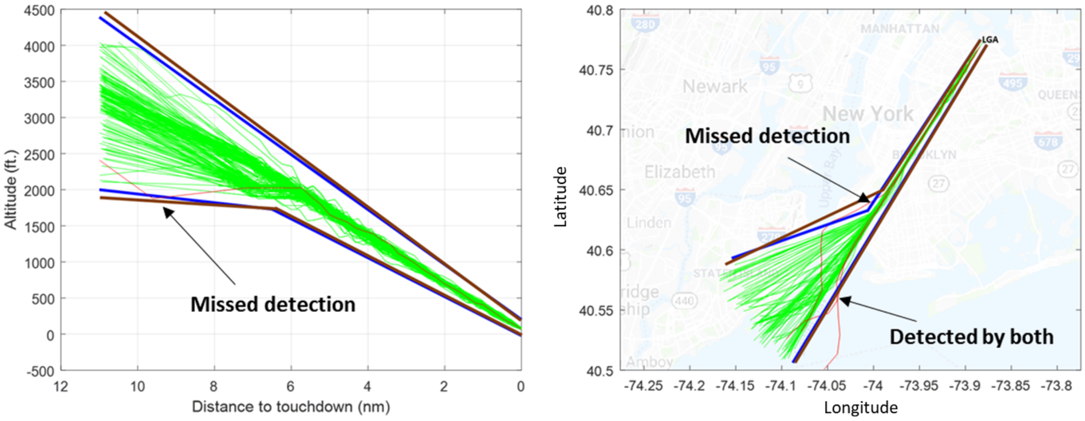


Figure 2.40. Illustrations of missed detection in vertical and horizontal dimensions: Blue model - Batch anomaly detection — Brown model – Online anomaly detection

the batch algorithm were analyzed. For this purpose, the degree of anomalousness is used, which is a metric for the severity of the anomaly. This degree is computed for each flight and each dimension and ranges between a value of 0 for a normal flight and 1 for the most anomalous flight. For all the anomalies that were missed detections, it was observed that they had a very low degree of anomalousness. Two representative examples are shown below in Fig. 2.40. Here, the blue model is the one generated by the batch test, while the brown model is generated by the online test.

It is observed that the missed anomalies barely violate the bound of the online monitoring model (brown): The degree of anomalousness for the vertical missed detection is 0.24, while the degree of anomalousness for the horizontal missed detection is 0.19. Thus, it is expected that some fine-tuning or human feedback can even further reduce the number of missed detections and make the online algorithm more effective.

3. PRECURSOR DETECTION IN TERMINAL AIRSPACE OPERATIONS

The objective of the research presented in this chapter is to develop a precursor detection algorithm based on supervised machine learning to augment the unsupervised anomaly detection algorithm developed in Chapter 2. The proposed precursor detection algorithm, called reactive TempAD, detects precursors using the surveillance data that correlate to specific anomalies in terminal airspace operations.

3.1 Preliminaries

The overarching framework of the proposed precursor detection algorithm is presented in Fig. 3.1 [52], which takes the time-series surveillance dataset as the input. With this, a labeled time-series dataset is constructed, comprising of features (or inputs) and labels (or outputs): a set of features which effectively represent every flight's behavior is constructed by extracting the information in the data; the labels representing which flight is normal or anomalous are obtained by an anomaly detection algorithm. The constructed dataset is fed into the proposed precursor detection algorithm, which is developed based on supervised learning techniques. The algorithm outputs precursor detection models which can be used to detect precursors corresponding to anomalous flights. In this section, the surveillance dataset, labeled dataset construction, the precursor detection algorithm and its performance measures are described in detail.

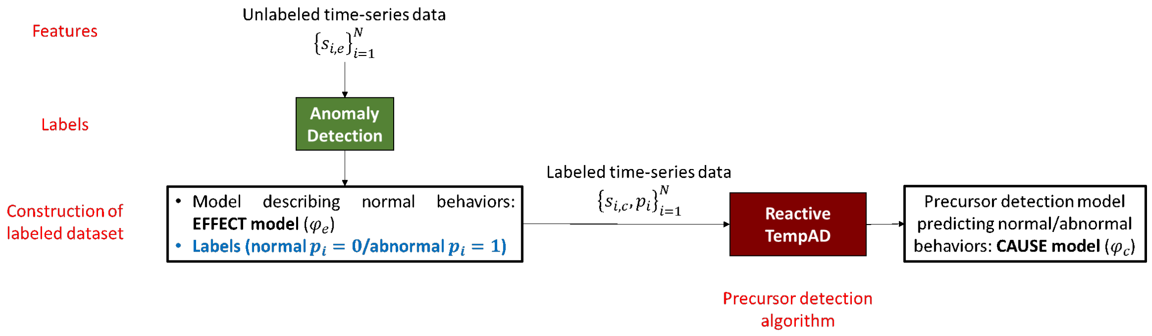

Features
Unlabeled time-series data
$\{s_{i,e}\}_{i=1}^{N}$
Labels
Anomaly Detection
Construction of labeled dataset
Model describing normal behaviors: EFFECT model (φ_e)
Labels (normal $p_i = 0$/abnormal $p_i = 1$)
Labeled time-series data
$\{s_{i,c}, p_i\}_{i=1}^{N}$
Reactive TempAD
Precursor detection model predicting normal/abnormal behaviors: CAUSE model (φ_c)
Precursor detection algorithm

Figure 3.1. Overarching framework of the proposed precursor detection algorithm

3.1.1 Surveillance dataset

The input to the proposed precursor detection algorithm is the real air traffic surveillance data, comprising of the aircraft track data which contains the aircraft states in a time-series format, such as the position (latitude and longitude), speed, altitude, heading with a unique flight identifier for each flight, recorded for arrival flights to the LaGuardia (LGA) airport. The surveillance data is sourced from the Airport Surface Detection Equipment – Model X (ASDE-X) and the Terminal Automation Information Service (TAIS). Both ASDE-X and TAIS are systems developed to record terminal airspace data to monitor the traffic around the airport. The ASDE-X dataset has a detection range of about 20 nautical miles from the airport and records data at every second. On the other hand, the TAIS dataset has a detection range of about 141 nautical miles from the airport with a rate of 5 seconds. The recording used in this chapter from ASDE-X is between April 6th to 24th, 2016, which has records of 9,634 arrivals at LGA, and from TAIS is during September, October and November 2016, which has records of 36,243 arrivals at LGA.

3.1.2 Construction of labeled dataset

From the real air traffic surveillance data, a labeled time-series dataset is constructed, which consists of a set of features (or inputs) and labels (or outputs). The features represent the behavior of a flight from which precursors can be detected, and the labels represent if the behavior of the flight is normal or abnormal.

Features

In the proposed precursor detection algorithm, the information in the real air traffic surveillance data is extracted to construct a set of features which can effectively identify the precursors for anomalies, which correspond to the following dimensions:

- Horizontal (H): Obtained from the positional (latitude and longitude) time-series data
- Vertical (V): Obtained from the altitude (h) time-series data
- Speed (S): Obtained from the ground speed (v) time-series data
- Specific Total Energy (STE): $h + v^2/2g$
- Specific Potential Energy Rate (SPER): $\dot{h}$
- Specific Kinetic Energy (SKE): $v^2/2g$
- Distances to preceding aircraft (d_V and d_H): Vertical and horizontal distances, respectively, to the nearest preceding flight (arriving or departing)

where g is the gravitational acceleration constant. Note that the first three dimensions are directly available from the ASDE-X and TAIS datasets, but the features in the energy dimensions and the distances to preceding aircraft need to be derived from the others. The energy features correspond to energy management, which is particularly important in the terminal airspace for detecting energy excess or deficient approaches [30]. The last feature attempts to capture airspace operations better by learning from loss of separation in the terminal airspace.

Labels

With the above m features ($m = 7$ here), let $\{s_i\}_{i=1}^N$ denote a set of N signals (or flights) where s_i consists of m time-series features, each of which has the length of T time steps, i.e., $\{s_i\}_{i=1}^N \in \mathbb{R}^{m \times T}$. The labels representing which flight is normal or abnormal are generated using the base-learning algorithm presented in Subsection 2.2.2. If the i^{th} flight violates the generated anomaly detection model, then it is labeled as anomalous; otherwise, normal.

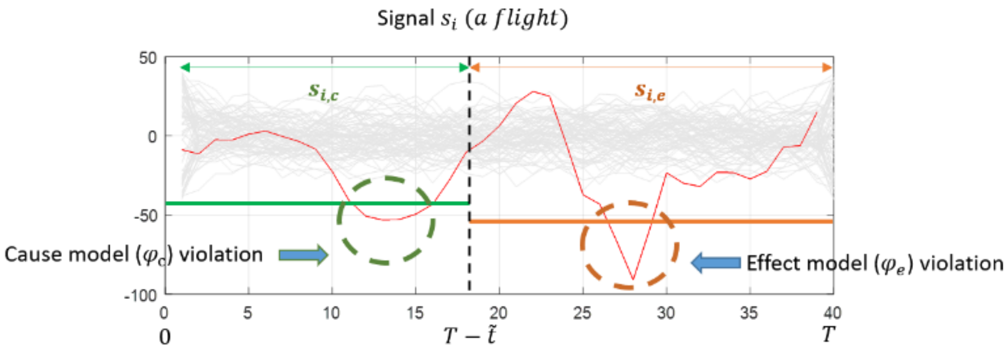


Figure 3.2. Cause and effect in a signal

Construction of labeled time-series dataset

Based on domain knowledge about air traffic operations in terminal airspace, the detected anomalies can be grouped into *anomaly types* by their characteristics, such as go-around, S-turn (path stretch), overspeed/underspeed, late interception of glideslope, energy excess/deficit, and change of runway anomalies [53]. To effectively detect precursors, one anomaly type are considered at a time, i.e., precursor detection models are generated for each anomaly type. In this regard, for a specific anomaly type of interest, the flights corresponding to the anomaly type are labeled as abnormal ($p_i = 1$) and all the other flights (even with other anomaly types) as normal ($p_i = 0$).

To detect a precursor for an anomaly, correlations in the form of cause-and-effect – called causal relations [41,54] – are identified in the signals by separating them into two parts: for s_i, let $s_{i,e}$ denote the latter part of s_i, for the interval $[T-\tilde{t}, T]$, and $s_{i,c}$ the earlier part, for the interval $[0, T-\tilde{t})$, as shown in Fig. 3.2. It can be assumed that an anomaly occurs in $s_{i,e}$ – the *effect signal* – and hence its corresponding precursor can be detected in $s_{i,c}$ – the *cause signal.* Note that $\tilde{t}$ is a user-chosen design parameter which can be tuned through analyzing the results of the anomaly detection algorithm (e.g., if all the anomalies occur during the final 5 time steps, $\tilde{t}$ is set as 5).

Therefore, from the unlabeled set of signals $\{s_i\}_{i=1}^N$, a labeled time-series dataset $\{s_{i,c}, p_i\}_{i=1}^N$ can be constructed for a given anomaly type of interest.

3.2 Algorithm development

The labeled dataset constructed above enables us to use supervised learning techniques. In general, for a given labeled set of N datapoints $\{(x_i, y_i)\}_{i=1}^N$ where x_i is the feature vector of the i^{th} datapoint and y_i is its label, a supervised learning algorithm attempts to find a function $g\colon X \to Y$ where $X \in \mathbb{R}^{m\times 1}$ is the input space and $Y \in \{0,1\}$ is the binary output space. For a decision variable θ (which may vary depending on the specific algorithm), a supervised learning algorithm solves an optimization problem,

$$\min_{\theta} \sum_{i=1}^{N} \text{loss}\left(g(x_i), y_i; \theta\right)$$

where $g(x_i)$ is a predicted label of i^{th} datapoint, y_i is its true label, and loss $(g(x_i), y_i)$ represents its classification error, or *loss*, computed by comparing the predicted and true labels, e.g., using Euclidean distance. Note that, in addition to the loss function, the optimization may also comprise of other objectives to minimize overfitting (regularization), maximize the classification margin, etc., depending on the specific technique used.

For detecting precursors from the constructed labeled time-series dataset $\{s_{i,c}, p_i\}_{i=1}^N$, the decision variable is a cause model φ_c that represents the bounds for the cause signals $\{s_{i,c}\}_{i=1}^N$ of normal flights ($p_i = 0$). In order to obtain φ_c by applying supervised learning techniques as described above, it is required to reduce the dimension of $s_{i,c} \in \mathbb{R}^{m\times\left(T-\tilde{t}\right)}$ to the dimension of the input space, $\mathbb{R}^{m\times 1}$. The underlying idea is that each time-series feature in $s_{i,c}$ can be effectively represented by a single-valued signed distance of cause signal $s_{i,c}$ to cause model φ_c. This signed distance, called *robustness degree* [54], should satisfy the property that if and only if $s_{i,c}$ violates φ_c at any time step, it is guaranteed that $p_i = 1$ (abnormal).

The supervised learning tool required to generate the precursor detection model can be of any kind, ranging from simple techniques such as regression and the K-nearest neighbor algorithm to complex ones such as the support vector machine (SVM), neural networks (NN) and random forest algorithms, based on the desired accuracy, robustness and performance, while considering the computational complexity. In this chapter, the SVM and the artificial neural network (ANN) algorithms are used for classifying the cause signals into anomalous and normal flights. Both of these algorithms are widely-used supervised learning algorithms which are robust to noise, are computationally inexpensive to run and deliver a good classification performance owing to their non-linear nature.

Furthermore, note that the cause model and the effect model may be implemented using different features. This is essential for precursor detection in flights, since a cause in one feature dimension can have an effect in another feature dimension. For example, a go-around anomaly could be detected using an effect model implemented using the altitude feature, but the cause model for precursor detection does not necessarily indicate any physical and operational significance in the altitude feature but can be better described in other features (e.g., energy), or even in a complex mathematical combination of several features.

3.3 Performance measures

In this section, the performance measures of the proposed precursor detection algorithm are presented, derived by analysis through two perspectives.

3.3.1 Classification performance

To determine how well the proposed precursor detection algorithm classifies the cause signals into normal and anomalous flights, the classification performance of the supervised learning algorithm is measured. This performance corresponds to the prediction capability of the precursors to anomalous flights. Using the accuracy (i.e., ratio of correctly labeled instances to all the instances) of the classifier is not an effective metric for rare events like anomalous flights, as the data is heavily imbalanced, i.e., the number of anomalies are extremely small compared to the number of normal observations (e.g., if there is 1% of anomalies in the dataset, wrongly classifying all the data points as 'normal' results in the accuracy of 99%, but this has no meaningful implication for precursor detection). Thus, the precision, the recall, and the F1 classification score are used, which are characterized as ratios of four data subsets, which, in the context of this chapter are:

- True positives (TP): Anomalous flights whose precursors are successfully detected
- False positives (FP): Normal flights which are predicted as anomalous (false alarms)
- False negatives (FN): Anomalous flights for which no precursors are found (missed detections)
- True negatives (TN): Normal flights which are also not predicted to be anomalous

The precision, the recall and the F1 classification score are computed as:

$$\begin{aligned}
\text{Precision} &= \text{ratio of true anomalies over total precursor detections} = \frac{TP}{TP+FP} \\
\text{Recall} &= \text{ratio of anomalies detected by precursors over total true anomalies} \\
&= \frac{TP}{TP+FN} \\
\text{F1 score} &= 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}
\end{aligned}$$

A high precision relates to a low false alarms rate, while a high recall relates to a low missed detections rate. A high F1 classification score – which is the harmonic mean of the precision and the recall – implies a good correlation between the occurrence of the precursor and the occurrence of the anomaly [55]. These three metrics can take values ranging from 0 to 1, by construction.

3.3.2 Look-ahead time

Considering that a safety-critical complex system such as the air traffic management system which involves a critical factor of several intricate human-machine and human-human interactions, one good performance metric of the algorithm is achieved through maximizing the *look-ahead time*, which is defined as the difference between the time of occurrence of the precursor and that of the anomaly, representing how soon (or early) the warning precursor is detected preceding the anomaly. A longer look-ahead time ensures that there is more time for the pilots and the ATCs to make a decision regarding the anomaly.

In the next section, specific anomaly types are chosen, and the proposed precursor detection algorithm is applied to predict these anomaly types using the aircraft states prior to the occurrence of the anomaly.

3.4 Test and analysis of reactive TempAD algorithm

In this section, the proposed precursor detection algorithm (reactive TempAD) is applied to terminal airspace operations. Through extensive tests of the algorithm, is observed that some anomaly types have only a single precursor within the dataset recording, while other anomaly types may have a sequence of precursors, and thus, both of these cases are analyzed separately.

3.4.1 Single precursor (go-around anomaly)

A go-around anomaly is one where the pilot in the anomalous flight aborts the landing during the last few nautical miles before touchdown, climbs using a missed approach procedure, circles around and attempts landing again. A visualization of the horizontal trajectory during a go-around is illustrated in Fig. 3.3. A go-around anomaly is a severe anomaly, in the sense that it is detected in all the dimensions.

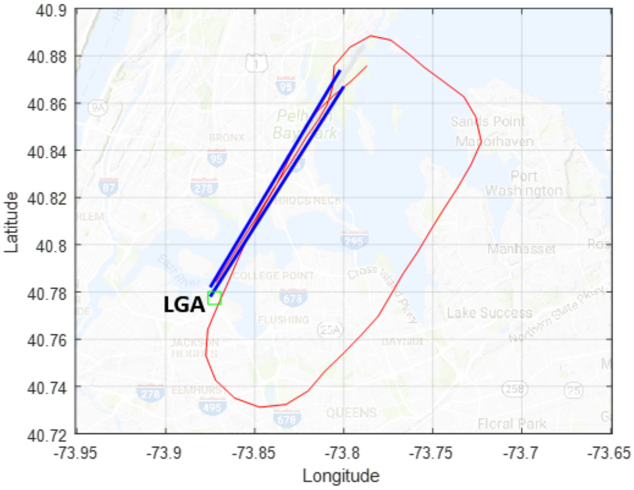


Figure 3.3. Horizontal view of go-around anomaly

The reactive TempAD is implemented to find a precursor for all the go-around anomalies detected at LGA airport across 19 days of the ASDE-X dataset (9,634 flights).

Reactive TempAD-SVM

In the first application, SVM is used as the supervised learning approach. To determine the precursor feature (the feature in which the precursor model is constructed) capable of delivering the best precursor detection performance, a search is performed across several candidate features. It was observed that a derived feature (f) from the energy features gave the highest F1 score from all tested candidate features:

$$f = \frac{SPER}{SPE} \times SKE = \frac{\dot{h}}{h} \times \frac{v^2}{g}$$

where SPER is the specific potential energy rate, SPE is the specific potential energy (or altitude) and SKE is the specific kinetic energy. Intuitively, this feature exhibits that a high sink rate ($\dot{h}$), a low altitude (h), or a high velocity (v) for a flight typically precedes the occurrence of a go-around anomaly.

The results of reactive TempAD with this feature f are presented in Figs. 3.4 and 3.5: the go-around flights (or signals) are in red and the normal flights are in green. Figures 3.4 and 3.5 demonstrate anomaly and precursor detection, respectively. Thus, if and only if the go-around flights violate the cause (or precursor) model (φ_c) in Fig. 3.5, it is guaranteed within a margin of error that the effect (or anomaly) model (φ_e) will be violated in Fig. 3.4, where the cause and effect models are respectively given as:

$$\begin{aligned} \varphi_c &= G_{[0,55)}\left(\frac{SPER}{SPE} \times SKE < 37\right) \\ \varphi_e &= G_{[55,60)}\left(1650 \times t + 75 \times altitude < 123750\right) \end{aligned} \tag{3.1}$$

Using the precursor detection model presented in Fig. 3.5, violations can be detected by determining signals that go above the precursor model (in blue) at any point

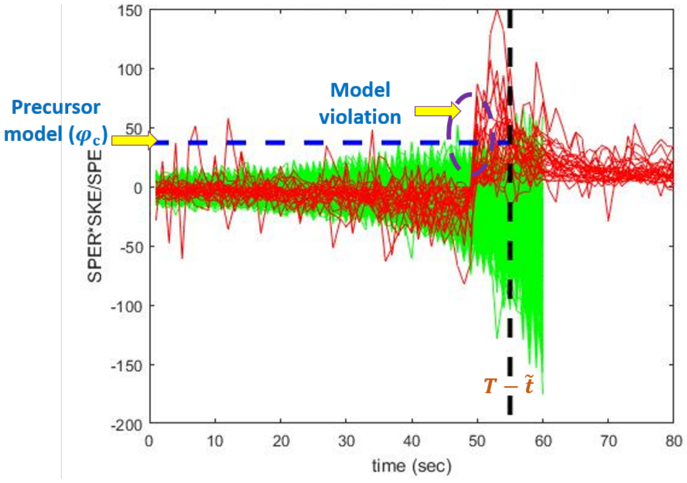


Figure 3.4. Go-around anomaly detection model

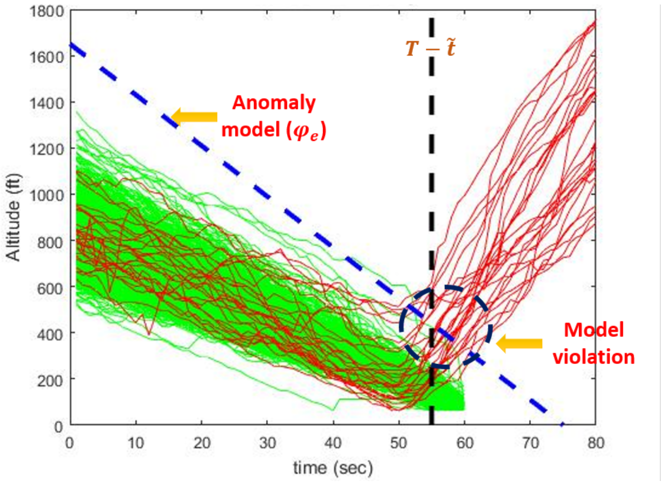


Figure 3.5. Go-around precursor detection model

Table 3.1. Confusion matrix for go-around precursor detection using reactive TempAD - SVM

Predicted **True**	**Anomalous flights**	**Normal Flights**	**Total**
Anomalous flights	True Positive 35	False negative 13	48
Normal flights	False positive 5	True negative 9,581	9,596
Total	40	9,594	9,634

of time through their trajectory in the first 55 time steps. Thus, using the models corresponding to Figures 3.4 and 3.5, true positives can be determined as flights that violate the precursor model in the first 55 time steps and the anomaly model in the last 5 time steps. Similarly, true negatives, false negatives, and false positives can be determined to form a confusion matrix, as shown in Table 3.1. The precision value is 0.875, the recall value is 0.729 and thus, the F1 score is 0.795. The average look-ahead time achieved using this feature f was 7 seconds. However, if it were to be applied to real-time implementation, it might not be sufficient enough since it gives the pilot and ATC only 7 seconds to react to the anomaly. From aircraft design and missed approach studies, 7-8 seconds are always required for engine spool-up during a go-around [56] and thus the look-ahead time achieved here is just feasible to be used.

Reactive TempAD-ANN

To improve the performance of the precursor detection algorithm, SVM is replaced with ANN. ANN automatically weighs the best combination of diverse features since it explores a more non-linear and complex hyperplane than SVM. Further, ANN can merge information contained in multiple features. Thus, it explores more complex causal relations than SVM and is thus expected to give a longer average look-ahead time along with a better classification performance. Note that due to the complex nature of the ANN algorithm, it is not always possible to visualize or to present the mathematical expression tractably, whereas the SVM algorithm results in a visualizable model, as in Eq. (3.1).

Using reactive TempAD-ANN, the results are improved with a precision of 0.891, a recall of 0.854 and an F1 classification score of 0.872, as shown in Table 3.2. More importantly, the average look-ahead time is increased to 11 seconds, an improvement of 57%.

It is important to note that the number of false negatives using the reactive TempAD-ANN has nearly halved from 13 (SVM) to 7 (ANN). Note that, among false

Table 3.2. Confusion matrix for go-around precursor detection using reactive TempAD - ANN

Predicted **True**	**Anomalous flights**	**Normal Flights**	**Total**
Anomalous flights	True Positive 41	False negative 7	48
Normal flights	False positive 5	True negative 9,581	9,596
Total	46	9,588	9,634

positives (false alarms) which detect a precursor but the flight is actually normal, and false negatives (missed detections) which detect no precursor but the flight is actually anomalous, the false negatives have a more crucial impact in safety-critical applications such as air traffic management, which demonstrates the usefulness of ANN over SVM, although physical intuition and visualization of the precursor detection model are lost.

3.4.2 Sequence of precursors (S-turn anomaly)

As shown in the study of the precursor to the go-around anomaly type, ANN improves the look-ahead time as well as the classification performance characterized by the F1 score. For the go-around case, only a single precursor could be found in the recorded data. On the other hand, for some anomaly types, it was observed that a sequence of precursors preceded the occurrence of the anomaly, which implies that there is a potential of increasing the look-ahead time than when just a single precursor is considered. Considering this observation, the implementation of the proposed algorithm was undertaken in such a manner that it presented the earliest precursor as the *true precursor*. Note that, the earliest precursor is called as the true precursor, since without the domain knowledge from a subject matter expert, it cannot be ascertained if the intermediate model violations are anomalies or precursors. For example, if a go-around anomaly is preceded by a loss of separation followed by a deficit energy approach, it is not always possible to determine if the energy violation was a consequence of directives from the ATC after a loss of separation is detected.

As an illustrative case, the study of the S-turn anomalies is presented, which are special anomalies in the horizontal dimension involving an extension of the flight path, also called a path stretch [51]. They are generally caused due to significant safety and operations related incidents and are critical anomalies in the terminal airspace [57]. Examples of S-turn trajectories detected by TempAD for arrivals to LGA runway 22 and runway 4 are presented in red in Fig. 3.6.

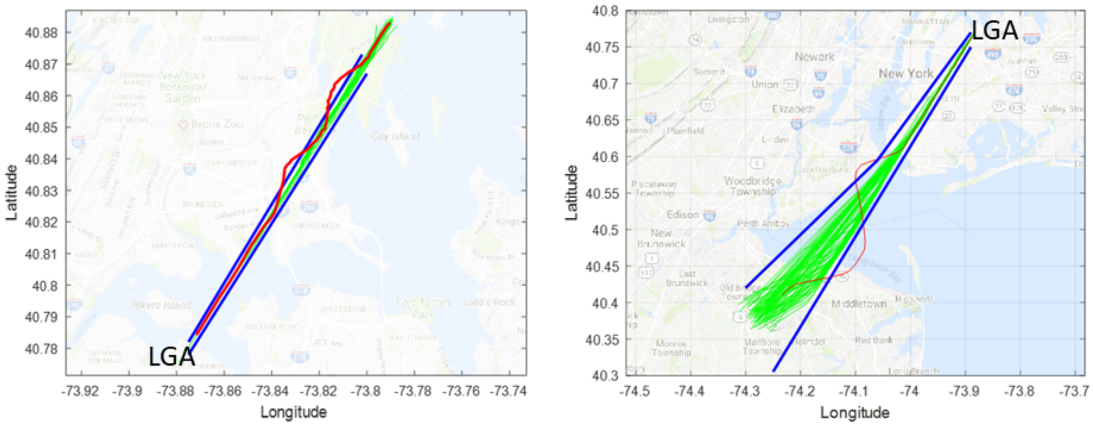


Figure 3.6. Horizontal view of S-turn anomaly in arrivals to RWY22 (left) and RWY4 (right) at LGA

Table 3.3. Confusion matrix for S-turn precursor detection using reactive TempAD - ANN

Predicted **True**	**Anomalous flights**	**Normal Flights**	**Total**
Anomalous flights	True Positive 738	False negative 171	909
Normal flights	False positive 91	True negative 35,243	35,334
Total	829	35,414	36,243

Reactive TempAD-ANN is implemented to find all precursors to S-turn anomalies detected at LGA airport across 88 days (September – November 2016) of the TAIS dataset (36,243 arrival flights). The most common reasons for the occurrence of an S-turn anomaly are to prevent loss-of-separation with a preceding flight or unstable approach in speed or altitude. Thus, the precursors for the S-turn anomaly were searched from candidate features comprising of separation from preceding flight (both arriving and departing flights) corresponding to the loss of separation cases; and approach speed and altitude, corresponding to the unstable approach cases. The results of testing reactive TempAD-ANN over 88 days of the TAIS dataset for arrivals to LGA are presented in Table 3.3.

The classification results have a precision of 0.89, a recall of 0.811 and thus, an F1 classification score of 0.849. To demonstrate that the proposed algorithm searches through the sequence of precursors that precede the S-turn anomaly in order to achieve a longer look-ahead time, an illustrative sequence is presented in Fig. 3.7. The flight has a loss of separation with a preceding flight, followed by a below normal glideslope. The flight intercepts the FAA-mandated glideslope only at 5 nautical miles from touchdown, as opposed to other flights that intercept the glideslope and start descending to the runway at 8 nautical miles from touchdown. This is followed

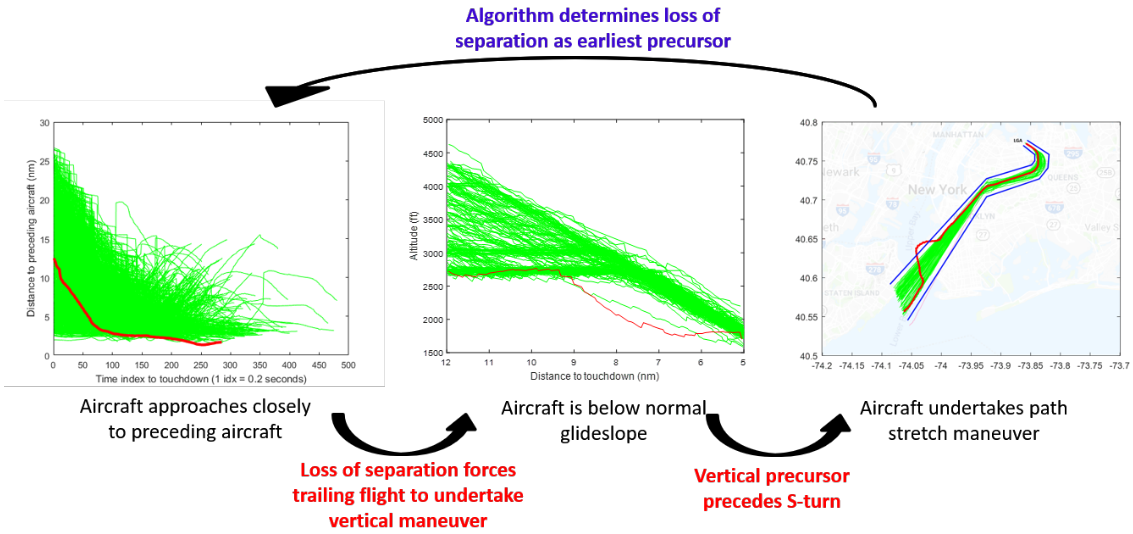


Figure 3.7. Sequence of precursors detected by the proposed algorithm for an anomalous flight arriving to RWY31 at LGA

by an S-turn anomaly before landing at runway 31 at LGA. If only the last precursor is considered (unstable approach; below normal glideslope), the look-ahead time is only 3 seconds, but if the earliest precursor (loss of separation) is considered, the look-ahead time is 14 seconds.

For the S-turn anomalies among the 36,243 arrivals at LGA, using only the single precursor approach resulted in an average look-ahead time of only 8.7 seconds. In contrast, the approach of analyzing the sequence of precursors delivers an average look-ahead time of 12.3 seconds, an improvement of 41%.

The results of applying the proposed precursor detection algorithm to anomalies within the real air traffic surveillance dataset presented in this section demonstrate how it can be used for predicting anomalies, and potentially be used as a decision-making tool to aid pilots and ATCs to improve the safety and efficiency of the terminal airspace system.

4. CONCLUSION

Concluding remarks and potential extensions of the research presented in this dissertation are elaborated upon in this chapter.

4.1 Discussion

This dissertation focuses on data-driven machine-learning applications in air traffic management (ATM), aimed at improving the safety and efficiency of operations in the terminal airspace of airports. This was achieved through two inter-related objectives: anomaly detection and precursor detection.

Anomaly detection in ATM gives air traffic controllers (ATCs), pilots, airlines, and policymakers insight into making air traffic operations safer and more efficient. Through this research, an event-based incremental learning anomaly detection algorithm dedicated to metroplex airspace operations was developed, which relies on unsupervised machine learning techniques. To make it effective in learning from the evolutionary behavior in the closely-coordinated metroplex airspace, the algorithm was made capable of ingesting both air traffic surveillance and airport operations datasets. The surveillance dataset allows the algorithm to learn from individual behaviors of each arriving or departing flight, while the operations dataset – comprising of vital information such as multi-airport coordinating policies, departure exit gate usage, operating runway, etc. – allows it to learn collective flight behaviors. The proposed algorithm assimilated these datasets together to obtain human-interpretable anomaly detection models, which effectively separated anomalous flights from normal flights. Furthermore, certain mathematical concepts embedded within the algorithm enabled human feedback, which can be used to improve the algorithm easily. To demonstrate the potential to be used in a real-time setting, the proposed algorithm was modified to ingest real-time flight data and monitor the occurrence of anomalies whenever they become evident in the aircraft states.

A conjugate problem to anomaly detection is *precursor detection*: to find a precedent for anomalies. In precursor detection, an algorithm was developed to search for specific events or occurrences that are found to typically precede the occurrence of an anomaly, such that it enabled the discovery of a causal relation with a degree of confidence, i.e., if a precursor (cause) is observed for a flight, the occurrence of the anomaly (effect) is reliably expected. Such a precursor detection algorithm was developed using supervised machine learning techniques, where the labels were generated using the proposed anomaly detection algorithm.

For the demonstration of both anomaly and precursor detection algorithms, extensive tests were performed using the real air traffic surveillance and airport operations datasets recorded in the New York metroplex area, comprising of arrival and departure operations at LaGuardia airport, John F. Kennedy International airport, and Newark Liberty International airport. The anomaly detection algorithm was shown to effectively capture the most common anomalies in the metroplex; furthermore, it was able to capture the evolution of the structure of the airspace over time and allowed the analysis of each anomaly. The augmented anomaly monitoring algorithm also demonstrated the potential to be used as a real-time application for ATM. The precursor detection algorithm was applied to two specific anomaly types: the go-around and S-turn anomaly types, which are typical conflict mitigation strategies in airports with high arrival traffic densities. For both these case studies, the precursor detection algorithm was shown to be capable of predicting most anomalies with sufficient warning time for pilots and ATCs.

Critical insight regarding ATM was gained from visualizations and analysis of the results of these extensive tests, which showed that the proposed algorithms have a potential to be used as decision-support tools that can aid pilots and ATCs to mitigate anomalies from ever occurring, thus improving the safety and efficiency of the metroplex airspace operations.

4.2 Future work

A deeper analysis of the anomalies and feedback by subject matter experts can assist in improving the detection capability of the algorithm. Furthermore, using a richer dataset – comprising of larger datasets recorded at other metroplexes – can improve the capability of the algorithm to detect more unusual anomalies, as it is expected that each metroplex presents a completely new environment and coordination strategies for the algorithm to learn. The effectiveness of the proposed anomaly detection algorithm, when applied to the terminal airspace, stimulates us to contemplate the application of the algorithms to detect surface anomalies and anomalies in the en-route airspace. An extension of the proposed precursor detection algorithm is to enable its applicability to several other types of anomalies detected in the terminal airspace, such as abnormal runway occupancy times, steep takeoff or approach, etc., and also to test the precursor detection algorithm for real-time operations.

REFERENCES

[1] IATA. Airlines' Perspective on Air Traffic Management, Jun 2019.

[2] Modernization of U.S. Airspace: NextGen, Sept 2019.

[3] SMART-NAS for Safe Trajectory Based Operations Project Description, Aug 2017.

[4] Clinton V Oster, John S Strong, and C Kurt Zorn. *Why airplanes crash: Aviation safety in a changing world*. Oxford University Press on Demand, 1992.

[5] Min Xue and Shannon Zelinski. Dynamic stochastic scheduler for integrated arrivals and departures. In *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, pages 1A2–1. IEEE, 2014.

[6] John Paul Clarke, Liling Ren, Evan McClain, David Schleicher, Sebastian Timar, Aditya Saraf, Donald Crisp, Richard Gutterud, Ryan Laroza, Terence Thompson, et al. Evaluating concepts for operations in metroplex terminal area airspace. *Journal of Aircraft*, 49(3):758–773, 2012.

[7] Santanu Das, Bryan L Matthews, Ashok N Srivastava, and Nikunj C Oza. 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*, pages 47–56. ACM, 2010.

[8] Lishuai Li, Maxime Gariel, R John Hansman, and Rafael Palacios. Anomaly detection in onboard-recorded flight data using cluster analysis. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pages 4A4–1. IEEE, 2011.

[9] Sriram Narasimhan and Lee Brownston. HyDE-A General Framework for Stochastic and Hybrid Model-based Diagnosis. *Proc. DX*, 7:162–169, 2007.

[10] Shun-Peng Zhu, Hong-Zhong Huang, Weiwen Peng, Hai-Kun Wang, and Sankaran Mahadevan. Probabilistic physics of failure-based framework for fatigue life prediction of aircraft gas turbine discs under uncertainty. *Reliability Engineering & System Safety*, 146:1–12, 2016.

[11] Shikha Agrawal and Jitendra Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713, 2015.

[12] Alek Gavrilovski, Hernando Jimenez, Dimitri N Mavris, Arjun H Rao, Sanghyun Shin, Inseok Hwang, and Karen Marais. Challenges and opportunities in flight data mining: A review of the state of the art. In *AIAA Infotech@ Aerospace*, page 0923, 2016.

[13] Eric A Wan. Neural network classification: A Bayesian interpretation. *IEEE Transactions on Neural Networks*, 1(4):303–305, 1990.

[14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[15] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. *GrC*, 5:718–721, 2005.

[16] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[17] Ali A Ghorbani, Wei Lu, and Mahbod Tavallaee. *Network intrusion detection and prevention: concepts and techniques*, volume 47. Springer Science & Business Media, 2009.

[18] Dimitry Gorinevsky, Bryan Matthews, and Rodney Martin. Aircraft anomaly detection using performance models trained on fleet data. In *Intelligent Data Understanding (CIDU), 2012 conference on*, pages 17–23. IEEE, 2012.

[19] Ashok N Srivastava. Greener aviation with virtual sensors: a case study. *Data Mining and Knowledge Discovery*, 24(2):443–471, 2012.

[20] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[22] Santanu Das, Bryan L Matthews, and Robert Lawrence. Fleet level anomaly detection of aviation safety data. In *Prognostics and Health Management (PHM), 2011 IEEE conference on*, pages 1–10. IEEE, 2011.

[23] Mohamed Cherif Dani, Cassiano Freixo, Francois-Xavier Jollois, and Mohamed Nadif. Unsupervised anomaly detection for aircraft condition monitoring system. In *Aerospace Conference, 2015 IEEE*, pages 1–7. IEEE, 2015.

[24] Sanghyun Shin and Inseok Hwang. Data-mining-based computer vision analytics for automated helicopter flight state inference. *Journal of Aerospace Information Systems*, pages 652–662, 2017.

[25] Lishuai Li, R John Hansman, Rafael Palacios, and Roy Welsch. Anomaly detection via a gaussian mixture model for flight operation and safety monitoring. *Transportation Research Part C: Emerging Technologies*, 64:45–57, 2016.

[26] Abhishek Vaidya, Sangjin Lee, and Inseok Hwang. Data-driven modeling and analysis framework for cockpit human–machine interaction issues. *Journal of Aerospace Information Systems*, pages 370–380, 2016.

[27] Lishuai Li, Santanu Das, R John Hansman, Rafael Palacios, and Ashok N Srivastava. Analysis of flight data using clustering techniques for detecting abnormal operations. *Journal of Aerospace Information Systems*, 12(9):587–598, 2015.

[28] Junshui Ma and Simon Perkins. Time-series novelty detection using one-class support vector machines. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 1741–1745. IEEE, 2003.

[29] Mark Schwabacher, Nikunj Oza, and Bryan Matthews. Unsupervised anomaly detection for liquid-fueled rocket propulsion health monitoring. *Journal of Aerospace Computing, Information, and Communication*, 6(7):464–482, 2009.

[30] Tejas G Puranik and Dimitri N Mavris. Anomaly detection in general-aviation operations using energy metrics and flight-data records. *Journal of Aerospace Information Systems*, pages 1–14, 2017.

[31] Irem Tumer and Anupa Bajwa. A survey of aircraft engine health monitoring systems. In *35th Joint Propulsion Conference and Exhibit*, page 2528, 1999.

[32] Wieslaw Staszewski, Geoffrey Tomlinson, Christian Boller, and Geof Tomlinson. *Health Monitoring of Aerospace Structures.* Wiley Online Library, 2004.

[33] Chze Eng Seah, Alinda Aligawesa, and Inseok Hwang. Algorithm for conformance monitoring in air traffic control. *Journal of Guidance, Control, and Dynamics*, 33(2):500–509, 2010.

[34] Garrett W Mann and Inseok Hwang. Four-Dimensional Aircraft Taxiway Conformance Monitoring with Constrained Stochastic Linear Hybrid Systems. *Journal of Guidance, Control, and Dynamics*, 35(5):1593–1604, 2012.

[35] Irem Y Tumer and Edward M Huff. Analysis of triaxial vibration data for health monitoring of helicopter gearboxes. *Journal of vibration and acoustics*, 125(1):120–128, 2003.

[36] HEG Powrie and CE Fisher. Engine health monitoring: towards total prognostics. In *1999 IEEE Aerospace Conference. Proceedings (Cat. No. 99TH8403)*, volume 3, pages 11–20. IEEE, 1999.

[37] Les Atlas, George Bloor, Tom Brotherton, L Howard, L Jaw, G Kacprzynski, G Karsai, R Mackey, J Mesick, R Reuter, et al. An evolvable tri-reasoner IVHM system. In *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*, volume 6, pages 3023–3037. IEEE, 2001.

[38] Vijay Manikandan Janakiraman, Bryan Matthews, and Nikunj Oza. Discovery of precursors to adverse events using time series data. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 639–647. SIAM, 2016.

[39] Prasad Naldurg, Koushik Sen, and Prasanna Thati. A temporal logic based framework for intrusion detection. In *International Conference on Formal Techniques for Networked and Distributed Systems*, pages 359–376. Springer, 2004.

[40] Austin Jones, Zhaodan Kong, and Calin Belta. Anomaly detection in cyber-physical systems: A formal methods approach. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 848–853. IEEE, 2014.

[41] Zhaodan Kong, Austin Jones, and Calin Belta. Temporal logics for learning and detection of anomalous behavior. *IEEE Transactions on Automatic Control*, 62(3):1210–1222, 2017.

[42] Airport Surface Detection Equipment, Model X (ASDE-X), Aug 2014.

[43] SWIM Terminal Data Distribution System (STDDS), Oct 2017.

[44] Aviation System Performance Metrics (ASPM) Web Data System, May 2019.

[45] New York ARTCC – LGA SOP, Nov 2018.

[46] New York ARTCC – JFK SOP, Nov 2018.

[47] New York ARTCC – EWR SOP, Nov 2018.

[48] Advisory Circular FAA. 120-71B. *Standard Operating Procedures and Pilot Monitoring Duties for Flight Deck Crewmembers*, 2017.

[49] Vanessa Gómez-Verdejo, Jerónimo Arenas-García, Miguel Lazaro-Gredilla, and Ángel Navia-Vázquez. Adaptive one-class support vector machine. *IEEE Transactions on Signal Processing*, 59(6):2975–2981, 2011.

[50] Raj Deshmukh and Inseok Hwang. Anomaly detection using temporal logic based learning for terminal airspace operations. In *AIAA Scitech 2019 Forum*, page 0682, 2019.

[51] Sebastian D Timar, Katy Griffin, Sherry Borener, and CJ Knickerbocker. Analysis of s-turn approaches at John F. Kennedy airport. In *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, pages 3C1–1. IEEE, 2012.

[52] Raj Deshmukh, Dawei Sun, and Inseok Hwang. Data-driven precursor detection algorithm for terminal airspace operations. In *13th USA/Europe Air Traffic Management R&D Seminar*, 2019.

[53] Raj Deshmukh and Inseok Hwang. Incremental-learning-based unsupervised anomaly detection algorithm for terminal airspace operations. *Journal of Aerospace Information Systems*, 16(9):362–384, 2019.

[54] Zhaodan Kong, Austin Jones, Ana Medina Ayala, Ebru Aydin Gol, and Calin Belta. Temporal logic inference for classification and prediction from data. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, pages 273–282. ACM, 2014.

[55] Yutaka Sasaki et al. The truth of the F-measure. *Teach Tutor mater*, 1(5):1–5, 2007.

[56] Tzvetomir Blajev and William Cutris. Go-Around Decision Making and Execution Project: Final Report to Flight Safety Foundation. *Flight Safety Foundation*, 2017.

[57] Steven Green, Robert Vivona, Michael Grace, and Tsung-Chou Fang. Field evaluation of descent Advisor Trajectory Prediction accuracy for en-route clearance advisories. In *Guidance, Navigation, and Control Conference and Exhibit*, page 4479, 1998.