Machine learning models for online anomaly detection in flight operations

Lucas Coelho e Silva* and Mayara Condé Rocha Murça†
Aeronautics Institute of Technology, São José dos Campos, São Paulo, 12228-900

Anomaly detection in flight operations data is a prominent approach to delivering actionable information for improved aviation performance as anomalies are often related to critical safety events or inefficient operations. In particular, the online detection of anomalies from streaming data provides the basis for novel real-time monitoring and alerting tools that can lead to more proactive and efficient decision-making during tactical operations. In this paper, we develop machine learning models for the online detection of flight trajectory anomalies from aircraft surveillance data. We focus on a specific type of operationally significant anomaly: go-around maneuvers. Air traffic controllers typically have to react quickly upon the occurrence of a go-around in order to reintegrate the aircraft into the landing sequence and maintain a safe and expeditious traffic flow. We develop a data-driven approach to anticipate such anomalous events for real-time air traffic control decision support. For this, two models are learned and compared in terms of predictive and computational performance: a Gaussian Mixture Model and a Temporal Convolutional Neural Network Model. The models are found to behave similarly, being able to anticipate 67% of the go-arounds with a false positive rate of less than 8%. Moreover, we find that the inclusion of weather features in both models allows for earlier prediction of go-arounds during the approach. In particular, the Gaussian Mixture Model is able to predict 78% of the go-around maneuvers correctly as early as 50 seconds from the runway.

I. Introduction

There has been a growing interest in anomaly detection initiatives within the flight operations context, driven mainly by the safety-oriented culture in aviation and the pursuit of improved operational efficiency in an increasingly complex, more autonomous, and evolved airspace. Previous research has mostly focused on the application of machine learning techniques for offline anomaly detection from complete data batches. While these efforts have boosted post-event analysis capabilities, driving the discovery of unmapped hazardous or inefficient operations, there is also a need for applications that enable the identification of anomalies in real-time. As anomalies can be precursors of unsafe and inefficient critical events, the early detection of anomalies from streaming data sources is valuable for improving response times and mitigating the impacts of such events. Moreover, online anomaly detection capabilities are enablers of real-time monitoring and alerting tools to allow for more proactive and efficient data-driven decision-making during tactical operations. This is particularly relevant when considering emerging Advanced Air Mobility (AAM)/Urban Air Mobility (UAM) operations and the future developments towards more flexible separation requirements as a way to enable scalability [1]. In this scenario, online anomaly detection is a valuable tool for ensuring safety levels. However, the online anomaly detection process brings several additional challenges related to the management of incomplete data, stringent targets on computing efficiency, and impacts of false alarms, among other factors.

In this paper, we develop and compare machine learning models for identifying operationally relevant anomalies in streaming flight operations data towards enabling the real-time detection of critical situations for air traffic management decision-making. We use actual aircraft surveillance data on a streaming schema to reproduce the operational scenario in which anomalous trajectory behaviors must be identified upon incomplete data continuously updated while the air traffic situation evolves. We particularly focus on trajectory anomalies associated with go-around maneuvers. A go-around is an event that typically requires air traffic controllers to make decisions reactively in order to reintegrate the aircraft into the landing sequence and maintain a safe and expeditious traffic flow. We develop two machine learning models to predict the occurrence of a go-around based on partial trajectory observations during the final approach phase: a Gaussian Mixture Model (GMM) and a Temporal Convolutional Neural Network (TCNN) model. For this, we rely on

*M.Sc. Student, Civil Engineering Department, Aeronautics Institute of Technology/coelho@ita.br.
†Professor, Civil Engineering Department, Aeronautics Institute of Technology/mayara@ita.br.

Automatic Dependent Surveillance-Broadcast (ADS-B) data for arriving flights at the Sao Paulo/Guarulhos International Airport (SBGR). In this sense, the main contribution of our work is the novel application of machine learning models for real-time anomaly detection in flight operations as well as the comparison of predictive and computational performance metrics and operational remarks between these models.

The structure of this paper is the following: Section II discusses the online anomaly detection problem in flight operations and reviews the related literature. Section III details the methodological approach, presenting the dataset used and the machine learning methods applied. Section IV presents the results and discussions, and Section V addresses the conclusions.

II. Background and literature review

The practice of finding valid and practically significant data structures inconsistent with a preconceived notion of expected normal behavior is known as anomaly detection. Within flight operations, the aviation subsystem that deals with the daily operations of flights, and in which the major stakeholders are aircraft operators and Air Traffic Management (ATM), anomaly detection initiatives are frequently linked to the investigation of safety or efficiency-related events [2]. In this sense, the objects of interest are often operational characteristics that affect system behavior, such as aircraft trajectories from the perspective of ATM; or, in the case of airlines, the identification of safety-related events and the disclosure of new dangers.

The anomaly detection process is often categorized in terms of the timeframe applicable to the statistical learning process and the model operation phase [2]. The modeling phase can take place offline by leveraging historical data on batch processing schemes, or it can happen online when the learning process updates the model as new data becomes available. In the latter, since the models are updated without the need to redeem the data used in the past, the online method is also referred to as incremental learning. In terms of the model operation, it is also distinguishable between the offline and online approaches. In offline model operation, the anomaly detection process takes place after the occurrence of the operations. On the contrary, an online model operation deals with an anomaly detection process that happens in real time, with streaming data as the operation progresses.

The approaches for formulating an anomaly detection problem are plentiful, and so are the algorithms and techniques suitable for the modeling process. Chandola et al. [3] presents an in-depth general review regarding the anomaly detection process while Coelho e Silva and Murça [2] present a data analytics framework applied to flight operations.

In the aviation domain, most anomaly detection formulations regard the offline model operation. Nevertheless, there are some initiatives in the literature that attempt to tackle online anomaly detection applications in flight operations. One of these lines of work pursues the offline identification of precursors to anomalous situations, with the goal of finding a cause model (precursor) such that if and only if the signal violates it, then it is guaranteed that it will exceed the normal flight threshold. In this sense, Deshmukh and Hwang [4] presents an unsupervised learning algorithm, named TempAD, that relies on temporal logic for anomaly detection in terminal airspace operations, with the generation of normal-flight parameter intervals that can be used in online settings. Subsequently, Deshmukh et al. [5] builds on the identified normal-flight parameter intervals and discusses procedures for identifying precursors for the detected anomalies in the terminal airspace operations surveillance data. Thereupon, the authors presented a supervised learning algorithm for precursor detection, Reactive TempAD, that could be used in online model operations.

Another line of work is that of leveraging probabilistic models, learned from offline data, applicable in online settings for prediction using partial observations. In this context, Murça [6] discusses anomaly detection via GMM, assessing the probabilities of new trajectories being generated by previously learned components, or trajectory patterns. Although used in an offline fashion within a framework for the characterization of air traffic flows based on flight trajectory data, this approach enables the prediction of developing, streaming trajectories, in the sense that it could be an alternative for online applications. Similarly, Murça and Oliveira [7] discusses the application of a GMM for creating probabilistic trajectory models for trajectory generation and prediction, while also envisioning online applications.

Finally, novel developments tackle online learning models that, albeit designed for offline model operation in the strict sense, may operate in a quasi-online fashion. Zhao et al. [8] discusses the application of GMM-based incremental clustering methods that can be updated with novel flight data, not requiring the execution of the clustering process ab initio. In this sense, it enables a quasi-online flight characterization process that could be used for anomaly detection. Furthermore, because it relies on GMM, it is possible to use the most recent cluster settings, updated online, for partial predictions on streaming trajectories.

Current literature still lacks anomaly detection applications focused on real-time model operation settings. In addition, with the proceeding progress of machine learning models and techniques, there are opportunities for exploring modeling approaches to problem settings that have not been thoroughly addressed, such as the online anomaly detection problem in flight operations. This work tackles these gaps with the novel development, application, and comparison of machine learning models for real-time anomaly detection in flight operations, as well as the discussion regarding the modeling process, the nature of the anomalies identified, the operational significance, and computational costs.

III. Methodological approach

In this paper, we learn statistical models for online anomaly detection with the application of machine learning techniques to flight tracking and meteorological data associated with arrival operations at SBGR. We leverage the anomaly detection framework in the flight operations domain proposed in our previous work [2] by adapting the base pipeline proposed in the framework. Figure 1 presents the workflow adopted in this paper.

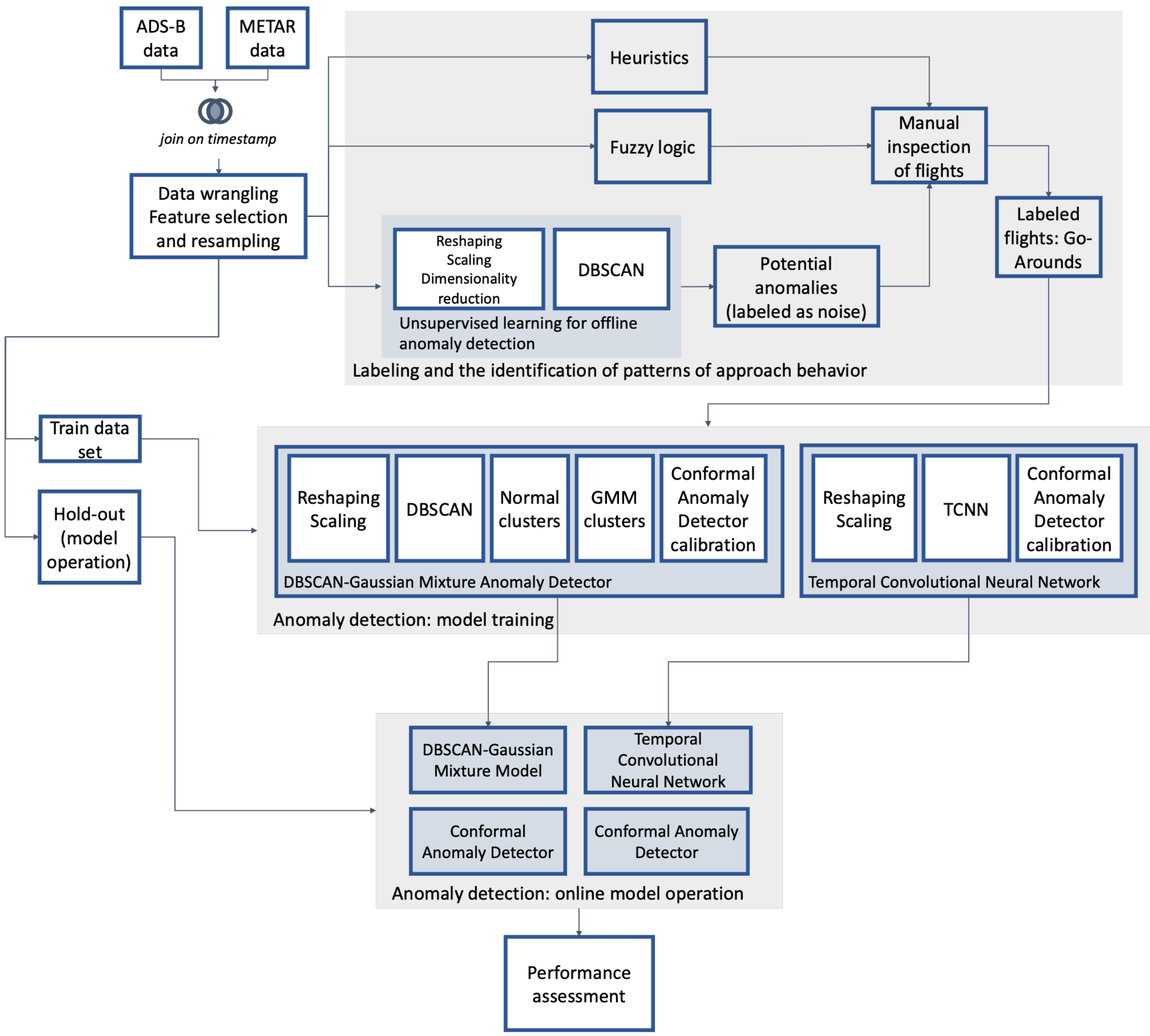


Fig. 1 Anomaly detection workflow: data processing and model conception.

After a baseline procedure of common data wrangling and pre-processing, we conduct the anomaly detection modeling procedure. For labeling every approach in terms of the occurrence of a go-around maneuver, we first perform an offline unsupervised learning process with the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [9] to discover potentially anomalous trajectories. In parallel, we process each approach according to heuristic rules while also subjecting each approach trajectory time series into a fuzzy logic go-around detection procedure.

For the online anomaly detection process, we learn two different machine learning models based on GMM and TCNN. The models are constructed with 80% of the data, available for model training, and evaluated on the remaining 20% hold-out data for replicating the live model operation. We construct and evaluate each model and draw comparisons regarding anomaly detection capabilities, computational efficiency, and operational remarks. For predicting a go-around maneuver based on partial trajectory observations derived from streaming data, we focus on the crossing point at which the aircraft is 1.5 NM away from the runway threshold. In a real operations scenario, this would be equivalent to a 30-second gap between the crossing point and the runway threshold in which air traffic controllers would be able to receive the information and prepare for it.

A. Dataset

The flight tracking dataset is gathered from the OpenSky Network and features $4,306$ arriving flights at SBGR during November 2019. It presents the flight information in a time series from which we extracted the following parameters and derived metrics for each flight:

- Record timestamp;
- Latitude;
- Longitude;
- Pressure altitude;
- Geometric altitude;
- Speed;
- Flight track;
- Specific Total Energy (STE);
- Specific Potential Energy Rate (SPER);
- Specific Kinetic Energy (SKE);
- Distance to the runway.

The meteorological dataset used in the modeling process features information regarding the presence of thunderstorms, flight rule regimes, winds, and gusts, gathered from the Meteorological Aerodrome Report (METAR).

Finally, to emulate the online model operation after training, we partition the dataset based on dates using the proportion 80%-20%. We use the first portion for training the anomaly detection models offline and the final 20% for replicating the online model usage.

B. Baseline procedures for model building

To ensure modeling on common data and assumptions, thus enabling the proper comparison of results, we designed a baseline procedure, applicable to all models, that guarantees model training on the same data subset. This involves the labeling process of the approach trajectories in terms of the occurrence of a go-around maneuver as well as the steps of feature construction, selection, and data resampling.

1. Data wrangling, feature selection, and trajectory resampling

For data preparation, we first restrict the flight trajectories to the portion of the data that happened within the terminal airspace. Then, when applicable, we split each flight trajectory into unique approach trajectories. This is because a single flight time series may contain information on two approach procedures – one before a go-around maneuver, and another one afterward, for example. In addition, we combine the trajectory dataset with the weather data according to the operation timestamp.

Finally, to account for the fact that each approach trajectory may present a different number of samples that, in addition, may be recorded at different portions of the procedure, we resample each trajectory via linear interpolation to present observations at every 0.5 NM after crossing the 8 NM mark from the runway threshold. This also relates to the need for representing the approach trajectories in the same dimensional space for the application of the selected machine learning models. Furthermore, resampling at fixed-distance intervals is convenient for model application in a real-world scenario, when a physical threshold may guide the prediction of the models when establishing a common procedure and policy.

2. Labeling of go-around anomalies

The model training process relies on the distinction between normal data and go-arounds. For that, the efforts for flagging potentially anomalous flights are threefold: a clustering process with DBSCAN, heuristic rules, and fuzzy logic. We perform unsupervised learning with the DBSCAN algorithm for discovering the main approach trajectory patterns and identifying potentially anomalous trajectories. In parallel, we also apply heuristic rules to the time series of geometric altitude and perform fuzzy-logic-based flight phase identification to flag approaches with go-around maneuvers. Finally, we visually inspect every anomalous flight flagged to ensure the presence of the target anomaly.

Unsupervised learning for go-around discovery: DBSCAN For the application of the DBSCAN pipeline, we reshape, scale, and perform dimensionality reduction before applying the clustering algorithm. Also, this DBSCAN pipeline is not to be confused with the one used by the GMM anomaly detector described in Section III.C.2. Although relying on the same sequence of steps, the former uses the entire dataset for labeling purposes, while the latter applies solely to the training data to avoid data leakage.

When reshaping each approach trajectory, the time series of latitude, longitude, geometric altitude, and speed are transformed into a single high-dimensional vector. For that, we rearrange the data interspersedly by representing each flight as a vector of the following shape in the high dimensional space, as per Equation 1:

$$x = [p_{0_{t_0}}, p_{1_{t_0}}, ..., p_{n_{t_0}}, ..., p_{0_{t_m}}, p_{1_{t_m}}, ..., p_{n_{t_m}}]. \tag{1}$$

where n is the number of parameters p and m is the number of time instances t.

Furthermore, we standardize the data with the removal of the mean and scaling to unit variance. Finally, we perform dimension reduction by applying a Principal Component Analysis (PCA) transformer, keeping 10 components of the data.

The DBSCAN algorithm relies on two principal hyperparameters: ϵ, which defines the distance threshold for considering two points as members of the same neighborhood during the clustering process, and MinPts, which refers to the minimum number of observations within the ϵ-neighborhood for considering a point as a core point during the process. We select an initial value for ϵ in accordance with Rahmah and Sitanggang [10] and adopt MinPts as 100 to prevent the generation of small clusters. Finally, we evaluate the clustering performance with the assessment of silhouettes [11] and identify any need to tune the ϵ hyperparameter further. The output of the DBSCAN pipeline is, first, trajectories labeled as normal and potential anomalies; second, the trajectory clusters to which each normal trajectory belongs. Here, we inspect the approaches labeled as noise (thus potentially anomalous) to confirm the occurrence of go-arounds.

3. Heuristics and fuzzy logic

A parallel effort for obtaining labels for approaches containing go-around maneuvers regards the application of heuristics and fuzzy logic. For heuristics, we mark every approach with a positive gain in geometric altitude as a potential anomaly. Regarding the application of fuzzy logic, we use the flight phase identifier algorithm proposed by Sun et al. [12] on each approach trajectory. Then, we mark every trajectory in which a climb phase is detected after a descent phase as a potential go-around anomaly. Similar to the potential anomalies identified by the DBSCAN pipeline, we visually inspect every flagged approach to confirm the anomalous characteristic of the trajectory.

C. Modeling approaches

1. Modeling process overview

To ensure a fair comparison between the two model categories explored in this paper, we conduct a training process with analogous assumptions and procedures. We construct two models for each category: a baseline that uses as features only the trajectory time series – latitude, longitude, geometric altitude, pressure altitude, SKE, STE, SPER, cumulative SKE, and cumulative STE; and an extended version that adds weather features – cross and aligned wind components, visibility, and the speed of wind gusts.

Moreover, in all scenarios, the identification of the go-around anomalous trajectories during online model operation build upon the construction of a Conformal Anomaly Detector (CAD) coupled with each model. We apply the CAD algorithm described by Laxhammar and Falkman [13] and Laxhammar and Falkman [14]. CAD relies on the definition of a Non-Conformity Measure (NCM) able to provide non-conformity scores whose frequencies are compared to a pre-defined desired significance level. In this paper, we use a Nearest-Neighbor (NN) algorithm with 10 neighbors as the NCM.

For selecting an appropriate value for the significance level, we assess the CAD's capability of identifying the available go-arounds in the training data according to a grid of possible significances between 0% and 5%. The smallest significance value at which the number of identified go-arounds become stable is adopted as the threshold for model operation. Then, for each new sample, we assess the proportion of the non-conformity scores of the reference points learned during the training step at least as large as the non-conformity score of the new sample, yielding a p-value that is checked against the desired significance.

2. Gaussian Mixture Model

The starting point of the GMM anomaly detector is a DBSCAN clustering pipeline analogous to the one defined in Section III.B.2, although applied to the training data only. Then, under the assumption that the trajectory distribution within each cluster can be modeled with Gaussian density functions, we model the identified trajectory clusters with a GMM. As mentioned in Section III.C.1, we construct two GMM models: a baseline, GMM-1, that uses features derived from the trajectory time series; and GMM-2, which, in addition, uses weather features.

Regarding the preprocessing steps of the input data, we reshape the approach series towards an interspersed, high-dimensional vector format, as defined by the Equation 1 of Section III.B.2. In the same fashion as the DBSCAN pipeline, we standardize the data with the removal of the mean and scaling to unit variance. For the GMM, however, dimension reduction is not performed.

For identifying anomalies in real time, we leverage GMM's ability to predict based on partial observations. We identify the go-around anomalies by calculating the conditional expectation of the approach trajectory, given the partial data. To assess the anomalous quality of the obtained trajectory, we fit a CAD during model training, coupled with a NN algorithm as the NCM for providing the nonconformity scores. Finally, there is a calibration step for selecting an appropriate value of the CAD's significance, enabling the calculation of go-around probabilities and identification of go-around anomalies during model operation. Figure 2 illustrates the prediction schematics during model operation.

Fig. 2 Model prediction schematics and anomaly detection workflow for the GMM anomaly detector.

In terms of software, we use the DBSCAN and Gaussian Mixture implementations of *scikit-learn* [15].

3. Temporal Convolutional Neural Network Model

For the construction of the neural network model, we use the open-source Python library *keras-tcn* [16], built on top of *Keras* [17], an Application Programming Interface (API) based on the machine learning platform *TensorFlow* [18]. Similarly to the GMM scenario, we construct two models, TCNN-1 and TCNN-2, with analogous assumptions to those of GMM-1 and GMM-2.

As specific pre-processing steps, instead of reshaping each approach trajectory series to a high-dimensional vector, the modeling data set is reshaped to constitute a tensor of dimensions p, s, and a, where p represents the flight parameters, such as latitude and speed; s denotes the instants at which the parameters are sampled, i.e., every 0.5 NM mark after 8 NM until the 1.5 NM threshold (for online prediction based on streaming data); and a indicates each independent approach trajectory. On the other hand, the target variable y is a matrix in which each row represents the geometric altitude series after the 1.5 NM mark. Figure 3 represents the feature set X and the target variable y for the TCNN model for a dataset containing n approach trajectories, k features (parameters), and j sampling instants of which i compose the observed X tensor.

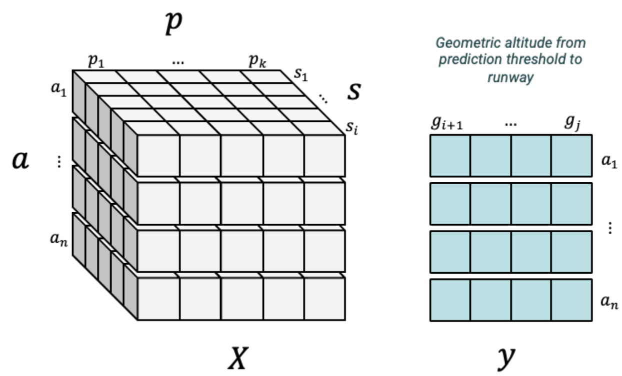


Fig. 3 Feature set X and target y for the TCNN model.

For training the neural networks, we used the Adam optimizer with a learning rate of 0.002 coupled with the Mean Squared Error (MSE) loss function. We train the network with a batch size of 10 for 10 epochs and also assess the validation loss in 20% of the training data. In terms of the neural network architecture, we configure the TCNN with 24 filters in the convolutional layers, a kernel size of 8, 1 stack in the residual blocks, and a dropout rate of 0.3, while also using skip connections from the input layer to each residual block.

The real-time identification of anomalies during model operation relies on the probability measures obtained from a CAD. For that, we fit the CAD coupled with a nearest-neighbor scorer on the predicted geometric altitude series for the normal approach trajectories available during the model training step, selecting an appropriate value for the significance hyperparameter that allows the detection of the go-around anomalies available in the train dataset. This procedure, analogous to the one described for the GMM in Section III.C.2, ensures a fair comparison between the two model categories. Figure 4 represents the prediction schema during model operation – after the model has been trained.

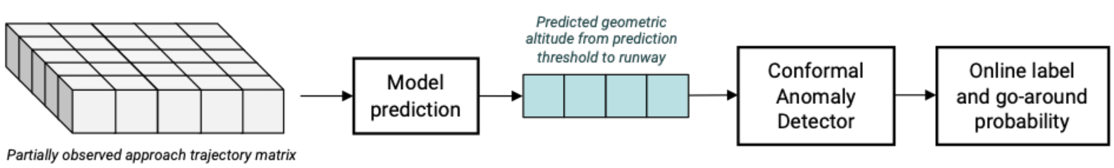


Fig. 4 Model prediction schematics and anomaly detection workflow for the TCNN anomaly detector.

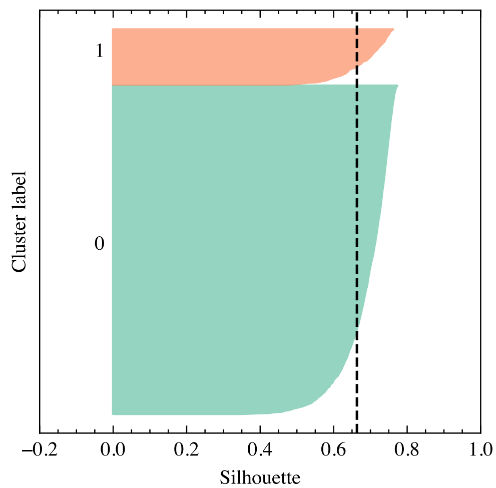
IV. Results and discussion

A. Identification of approach patterns and labeling of go-arounds

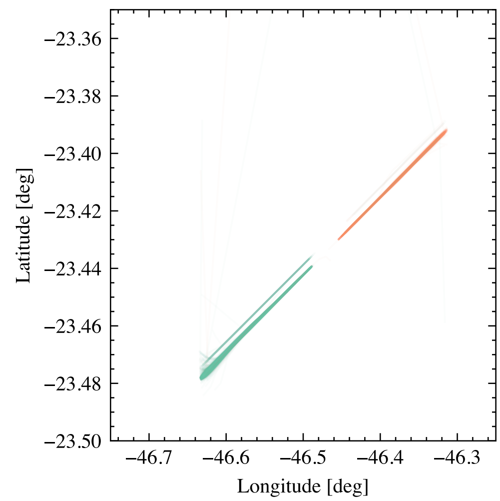
As we focus on a specific type of operational anomalies - go-around maneuvers - one of the first steps in the workflow consists of identifying these anomalies to allow the training process of the machine learning models on labeled data. As discussed in Section III.B.2, we conduct a labeling process based on clustering with a DBSCAN pipeline, heuristics, fuzzy logic, and a final visual inspection of the flagged approach trajectories.

1. Clustering of approach trajectories

As mentioned in Section III.B.2, before conducting the clustering process, we resample each trajectory, resulting in observations sampled at every 0.5 NM after crossing the 8 NM mark from the runway threshold. In addition, we perform dimension reduction with PCA, keeping 10 components of the data. We then apply the DBSCAN algorithm to filter out noisy and potentially anomalous trajectories and to learn the major clusters of normal behavior associated with the different runways used for landing. We identify two approach clusters within the trajectories, one for each runway. Figure 5a depicts the silhouette assessment of the two identified clusters, with an average value of 0.67, and Figure 5b displays the horizontal profile of the clustered approach trajectories.



(a) Silhouette plot for the clusters identified with DBSCAN.



(b) Clustered approach trajectories obtained via the clustering process with DBSCAN.

Fig. 5 DBSCAN clustering model results: silhouette and identified clusters.

2. Identified go-arounds

As a result of the labeling process, we have identified 38 flights out of the 4, 306 to contain go-around maneuvers. The small proportion of go-arounds within the approach dataset corroborates with the application of anomaly detection techniques in this scenario. Figure 6 depicts the vertical trajectories of the go-around anomalies identified within this dataset. Out of the 38 go-arounds, 29 belong to the training data and 9 remain in the hold-out subset used for model operation.

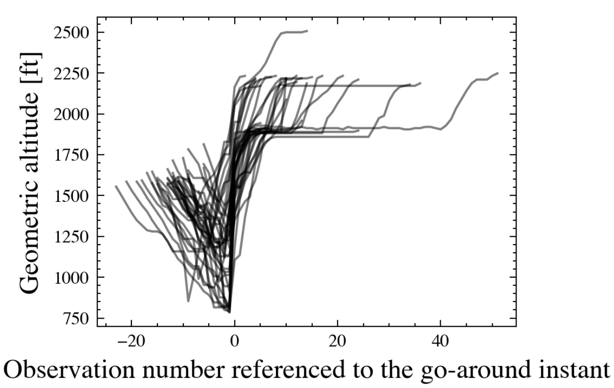


Fig. 6 Go-around maneuvers identified during the labeling process.

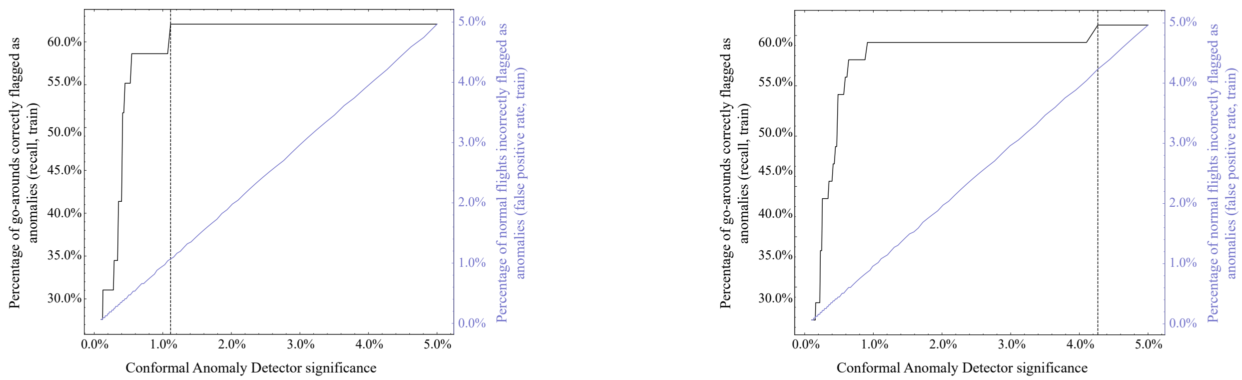
B. Anomaly detection model learning

1. Gaussian Mixture Model Anomaly Detector

The training process of the GMM models happens in an unsupervised fashion, in which the clusters unveiled by the coupled DBSCAN pipeline feed the construction of the mirrored Gaussian Mixture clusters, while the choice of the significance level for the CAD happens according to the procedure discussed in Section III.C.1.

Figure 7 presents the calibration curve for the GMM in terms of the percentage of go-arounds correctly identified as anomalous approaches and the false positive rates in the train dataset according to various possible values for the CAD significance hyperparameter. For GMM-1, the model without the weather data, a significance threshold of 1.11%

was sufficient for identifying over 60% of the go-arounds of the training set. For GMM-2, the required significance is higher (4.27%) for successfully flagging a similar amount of go-around approach trajectories. Section IV.C details the performance of GMM-1 and GMM-2 during the model operation phase.



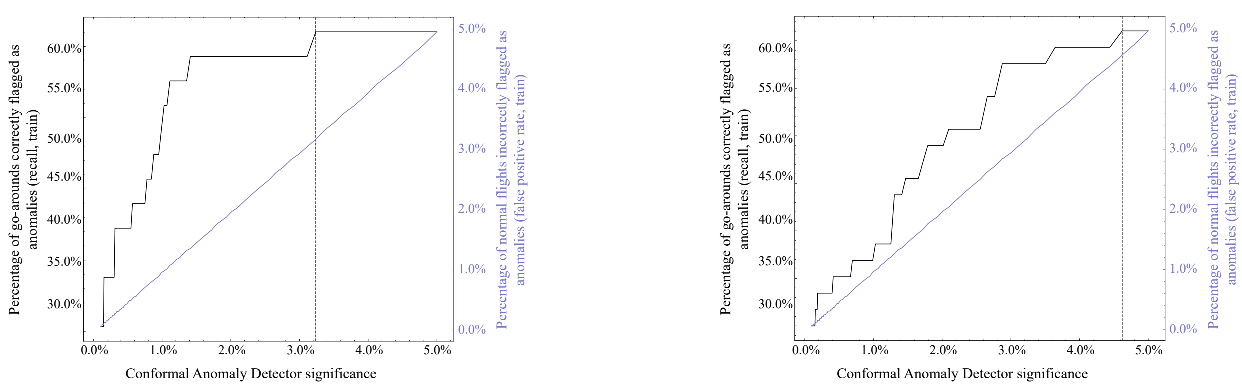
(a) GMM-1, with a selected significance of 1.11%.

(b) GMM-2, with a selected significance of 4.27%.

Fig. 7 Calibration curve for GMM: percentage of go-arounds correctly identified as anomalous approaches and false positive rates in the train dataset according to Conformal Anomaly Detector significance.

2. Temporal Convolutional Neural Network Anomaly Detector

For the TCNN models, we train the networks in terms of the MSE loss function, with a batch size of 10 for 10 epochs, with the assessment of the validation loss in 20% of the training data. The neural network architecture is the one described in Section III.C.3. Similar to the GMM scenario, Figure 8 presents the calibration curve for the TCNN models regarding the percentage of go-arounds correctly identified as anomalies and the false positive rates in the train dataset according to a range of candidate values for the CAD significance. TCNN-1, the TCNN model without the weather data, required a significance threshold of 3.24% for identifying over 60% of the go-arounds of the training set. For TCNN-2, on the other hand, the required significance for successfully flagging a similar amount of go-around approach trajectories is higher (4.62%). While TCNN-2 thresholds are close to GMM-2's threshold of 4.27%, TCNN-1's selected significance level is almost three times the significance level of GMM-1.



(a) TCNN-1, with a selected significance of 3.24%.

(b) TCNN-2, with a selected significance of 4.62%.

Fig. 8 Calibration curve for the TCNN models: percentage of go-arounds correctly identified as anomalous approaches and false positive rates in the train dataset according to Conformal Anomaly Detector significance.

Section IV.C compares the results from TCNN-1 and TCNN-2 during model operation.

C. Model operation

As mentioned in Section III.A, to define a hold-out, out-of-time subset for emulating the online model operation after training, we partition the dataset in approximately 80-20%. This 20% hold-out data refers to the flight operations from November 25, 2019, to November 30, 2019.

After replicating the streaming operation phase for each of the modeling strategies, we compare the models in terms of the proportion of anomalies, the identified anomalous trajectories themselves as well as the rate of false alarms. Figure 9 shows the percentage of approach trajectories flagged as anomalous by each model during the simulated model operations. While the go-around trajectories represent 0.84% of all trajectories evaluated during model operation, the models containing weather data flagged as much as 17.5% of the trajectories on November 27 as potentially anomalous. This is closely related to the false positive rate figures shown by Figure 10a, indicating that the majority of the flagged trajectories are indeed false positives. The degree to which normal trajectories are flagged as go-arounds, culminating in false positives, varies with the analyzed models, however. GMM-1 presents the lowest overall false positive rate: 4.4% when considering the entire model operation period, as shown in the summarizing overall numbers of Table 1. On the other hand, GMM-2 displays the highest false positive rate for the operation period. This performance "degradation" with the addition of the weather variables of the cross and aligned wind components, visibility, and wind gust speed may be explained by the fact that, while a GMM assumes a Gaussian process for data generation, that is often not the case for weather data. Wind gusts, for example, do not follow Gaussian densities [19], and are often modeled as Weibull distributions [20]. Nevertheless, the same pattern of performance "degradation" regarding false positives with the inclusion of weather data is observed in the TCNN models, although with less pronounced effects. TCNN-1, without the weather data, displays an overall false positive rate of 7.3%, while TCNN-2 falsely flags as anomalies 8% of the normal approach trajectories. Finally, we place "degradation" on quotes as it could be the case that the false positives are in fact anomalous trajectories other than go-arounds, such as unstable approaches. In this case, the inclusion of weather data would have allowed for the identification of these anomalies which we have chosen not to map in this study, making the further investigation of the flagged anomalies necessary.

The high variability in the false positive rates contrasts with the uniformity observed regarding the percentage of go-arounds correctly flagged as anomalies by each model, represented in Figure 10b. Except for November 27, when the weather-based models successfully identified every go-around trajectory, one more than GMM-1 and TCNN-1, the models presented identical performance. In both model categories, the addition of weather data improved recall and allowed for the detection of one extra go-around maneuver.

Finally, the similar performance metrics regarding the capability of detecting go-around approach trajectories in advance may indicate that further improving the detection performance, not at the expense of false positives by loosening the significance requirements, may relate more to the dataset rather than to the models themselves, in the sense that it may come from better features and wrangling techniques rather than specific modeling approaches.

Table 1 Summarizing metrics for the entire model operation period, per evaluated model.

	GMM-1	GMM-2	TCNN-1	TCNN-2
% of go-arounds correctly identified	67%	78%	67%	78%
False positive rate	4.4%	8.9%	7.3%	8.0%

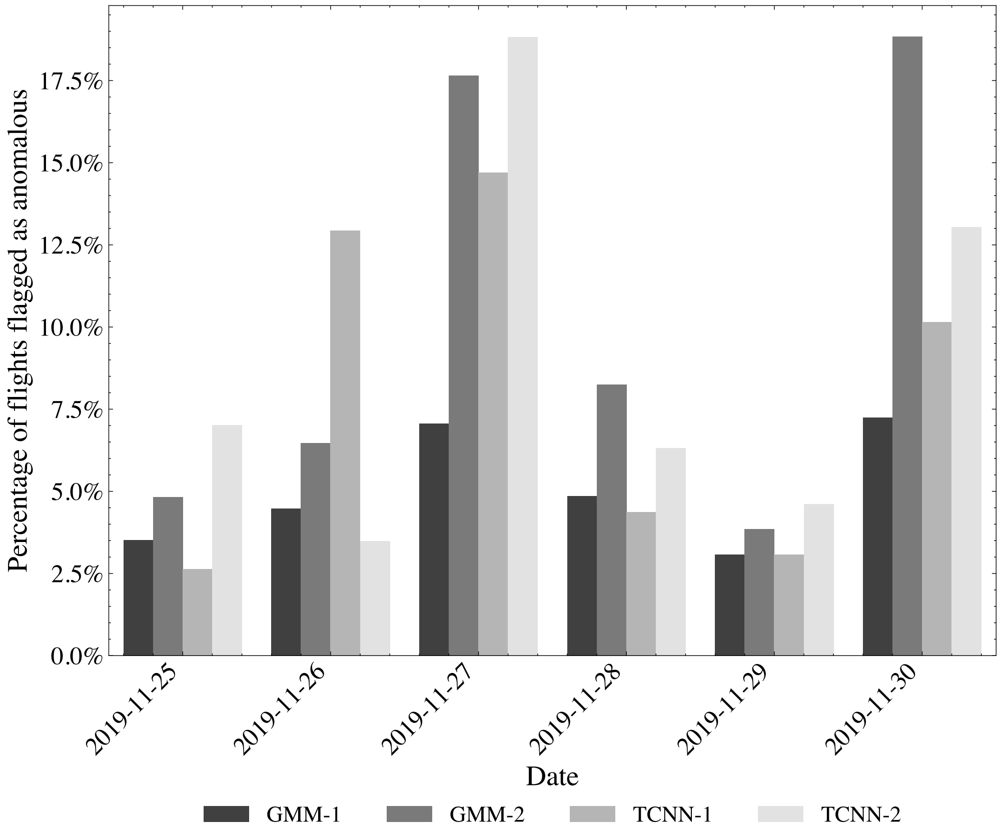
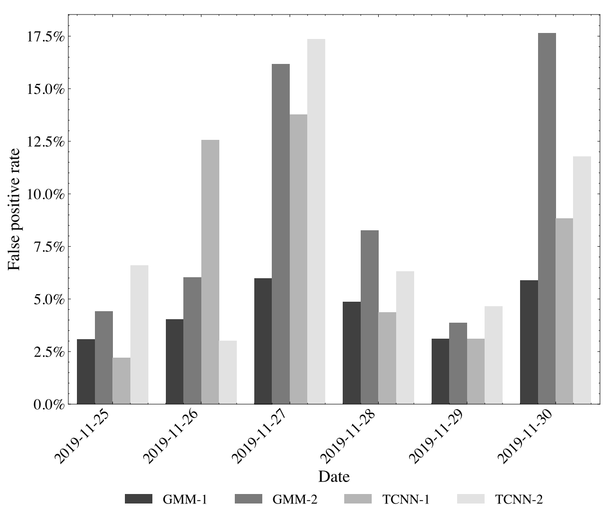
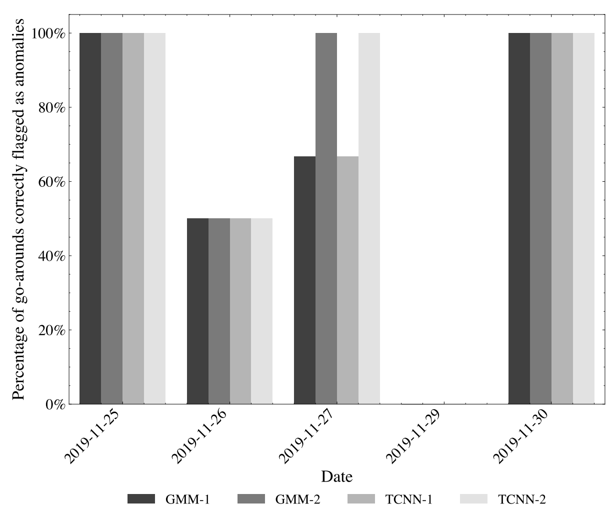


Fig. 9 Percentage of approach trajectories flagged as anomalies by each model during simulated model operation.



(a) Percentage of normal approach trajectories incorrectly flagged as anomalies by each model during simulated model operation, daily.



(b) Percentage of go-around maneuvers correctly flagged as anomalies by each model during simulated model operation, daily.

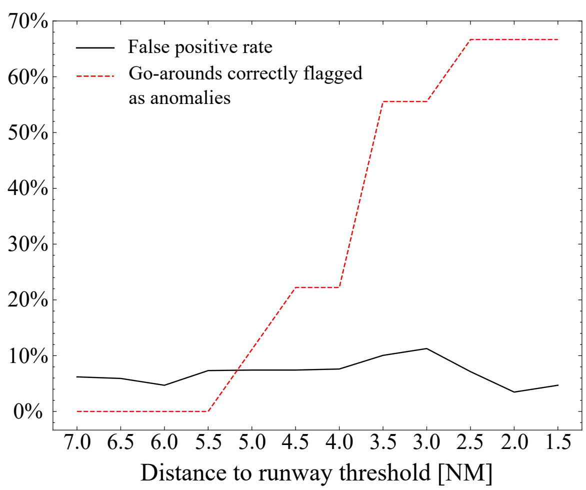
Fig. 10 False positive rate and percentage of go-around maneuvers correctly flagged as anomalies by each model during simulated model operation.

D. Model performance as a function of the distance to the runway

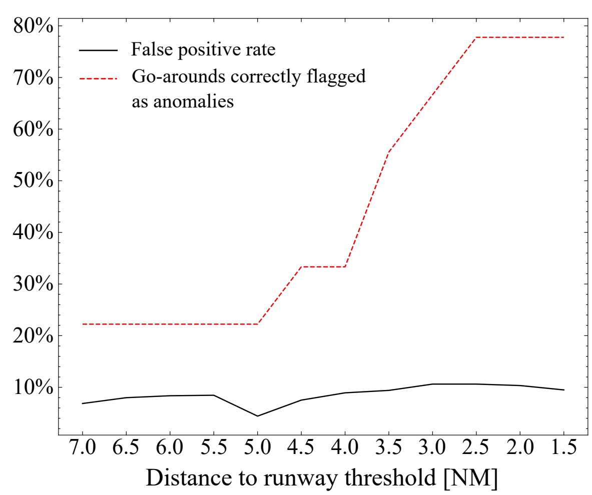
Another investigation performed is the one regarding the model performance in terms of the completeness of the data used for prediction, aiming at responding to the question of how early in the final approach trajectory it is viable to apply the models for anticipating the go-around maneuvers.

For this task, GMM poses a clear advantage in comparison to the TCNN models, since it is not necessary to fit a GMM model multiple times. As each model feature is treated independently and GMM allows for the prediction using partial data via the assessment of conditional expectations and marginal distributions, evaluating the performance as a function of how far the aircraft is from the runway threshold, hence data completeness, is straightforward. On the other hand, the TCNN models as we have formulated rely on fixed-shape feature set tensors and are trained to predict a specific number of observations for the geometric altitude series. In this sense, to assess model performance as a function of the distance to the runway, we trained different TCNN models, one for each prediction threshold. Finally, because the fitting of the CAD relies on the exhaustive search of a grid containing possible values for the significance parameters, the results presented in this investigation may differ slightly from the ones discussed in Section IV.C.

Figure 11b shows the model performance in terms of false positive rates and the proportion of go-around trajectories correctly flagged as anomalies for GMM-1 and GMM-2, while Figure 12 displays the equivalent information for TCNN-1 and TCNN-2. As expected, model performance increases with data completeness in every scenario. GMM-1 and GMM-2, however, stabilize at the 2.5 NM distance threshold, indicating the capability of predicting go-around maneuvers as early as 50 seconds from the runway, considering a fixed-speed approach of 180 kt. It is also clear that the inclusion of weather variables in GMM-2 and TCNN-2 allows for earlier detection of go-around trajectories, being able to capture 20% of the go-arounds as far as 7.0 NM away, in contrast with the models not based on weather, in which the first detection of go-arounds come at 5.5 NM and 6 NM for GMM-1 and TCNN-1, respectively. Lastly, it is also noticeable that the false positive rate does not change significantly with the distance to the runway, indicating the robustness of the CAD algorithm in establishing detection thresholds.

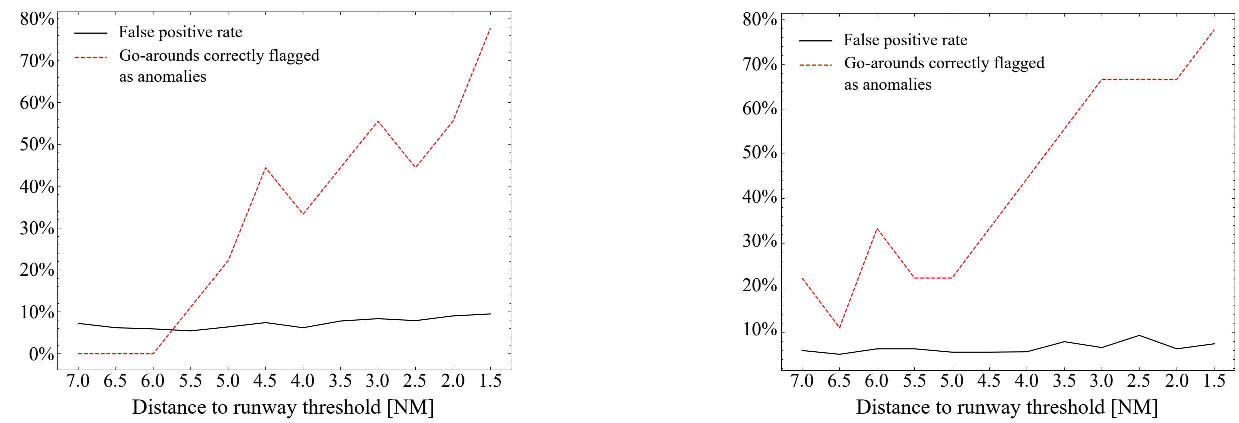


(a) GMM-1.



(b) GMM-2.

Fig. 11 False positive rate and percentage of go-around maneuvers correctly flagged as anomalies by the GMMs as a function of the distance to the runway threshold.



(a) TCNN-1. **(b) TCNN-2.**

Fig. 12 False positive rate and percentage of go-around maneuvers correctly flagged as anomalies by the TCNN models as a function of the distance to the runway threshold.

E. Model complexity and computational costs

In terms of computational costs, there is a clear distinction between the time required for training GMM-1 and GMM-2 and the time required to train the neural networks. While GMM-1 and GMM-2 averaged approximately 2.3 seconds each, TCNN-1 required on average 105 seconds for model training, while TCNN-2 took 90 seconds, on average. The training step of the CAD took, on average, 1.5 seconds per iteration across all analyzed models. These numbers refer to a training process conducted using Python 3.10.4 on a 2021 Apple MacBook Pro with an M1 Pro chip and 16 GB of RAM.

Regarding live operation performance, the time required for making a prediction for a single trajectory averaged close to 0.5 seconds for every model coupled with the CAD. Most of the computational cost in this phase regards the pairwise distance computation by the nearest-neighbor NCM coupled with the anomaly detector, as illustrated in Figure 13 The model prediction steps themselves took 0.02 seconds for the TCNN models and 0.002 seconds for GMM-1 and GMM-2, on average. The 0.5-second prediction time cost is enough for the airplane to travel approximately 0.025 NM (46 m) between receiving the data and getting a status regarding the probability of a go-around. For reducing this time, one can leverage the Inductive Conformal Anomaly Detector (ICAD) algorithm proposed by Laxhammar and Falkman [14].

Finally, these computational cost metrics are by no means exhaustive but rather serve as guidelines in terms of the differences in orders of magnitude and for envisioning possible impacts during model operation in a realistic operational setting.

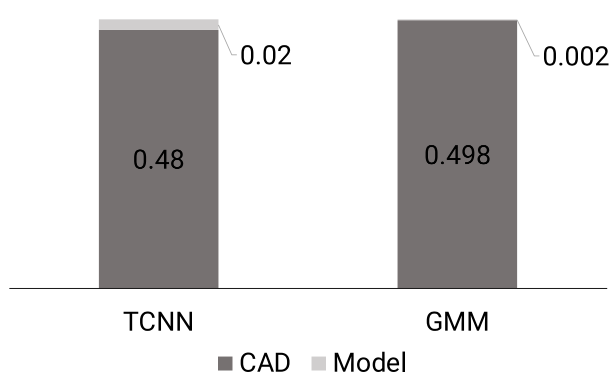


Fig. 13 Elapsed time in seconds for predicting whether a go-around maneuver will take place for a single approach trajectory.

V. Conclusion

In this paper, we explore the development, application, and comparison of machine learning models for real-time anomaly detection in flight operations. We focus on go-around maneuvers, a specific type of operationally significant anomaly that requires the prompt reaction of air traffic controllers in order to reintegrate the aircraft into the landing sequence and maintain a safe and expeditious traffic flow. In this sense, we provide a first step toward the applicability of machine-learning-based anomaly detection initiatives in real-time settings that would aid air traffic control.

We explore two different machine learning model categories: Gaussian Mixture Model (GMM) and Temporal Convolutional Neural Network (TCNN). For each model category, we build two models: one based solely on trajectory data (GMM-1 and TCNN-1), and the other that considers, in addition, weather data (GMM-2 and TCNN-2).

The models are found to behave similarly. GMM-1 and TCNN-1 are able to anticipate 67% of the go-arounds with a false positive rate of less than 8% at a distance of 1.5 NM from the runway. We observe that the inclusion of weather features improves the model's capability of detecting go-around maneuvers at the expense of an increase in the false positive rate. We have also analyzed the model performance in terms of the completeness of the data used for prediction, investigating how early in the final approach trajectory it is viable to apply the models for anticipating the go-around maneuvers. While model performance increases with data completeness in every scenario, GMM-1 and GMM-2 stabilize at the 2.5 NM distance threshold, indicating the capability of predicting go-around maneuvers with true positive rates of 67% and 78%, respectively, as early as 50 seconds from the runway. It is also clear that the inclusion of weather features allows for earlier detection of go-around trajectories during the approach, being able to anticipate 20% of the go-arounds as far as 7.0 NM away, in contrast with the models not based on weather, in which the first detection of go-arounds come at the 5.5 NM mark.

For future research, it can be worth assessing the performance of the models in different operational scenarios, such as varying weather conditions, airport layouts, or air traffic flow characteristics. Another natural further step is the analysis of a broader range of anomalies other than go-arounds. The inclusion of new features such as delays can also provide new insights and improve the model performance. Moreover, the investigation of different model categories, such as the Mixture of Generalized Lambda Distributions, may bring new understanding and serve as a suitable way to model the non-Gaussian processes present in flight operations and weather. Furthermore, the lack of explanatory information regarding the flagged anomalies may hinder the real-life application of the models. In this sense, online anomaly explanation and causal inference pose viable future research directions for enabling users to build trust in decision-support systems, thus easing the adoption process.

References

[1] Murça, M. C. R., "Identification and prediction of urban airspace availability for emerging air mobility operations," *Transportation Research Part C: Emerging Technologies*, Vol. 131, 2021, p. 103274. https://doi.org/10.1016/j.trc.2021.103274, URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X21002862.

[2] Coelho e Silva, L., and Murça, M. C. R., "A data analytics framework for anomaly detection in flight operations," *Journal of Air Transport Management*, Vol. 110, 2023, p. 102409. https://doi.org/https://doi.org/10.1016/j.jairtraman.2023.102409, URL https://www.sciencedirect.com/science/article/pii/S0969699723000522.

[3] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly detection: A survey," *ACM Computing Surveys*, Vol. 41, No. 3, 2009, p. 58. https://doi.org/10.1145/1541880.1541882, URL http://doi.acm.org/10.1145/1541880.1541882.

[4] Deshmukh, R., and Hwang, I., "Anomaly Detection Using Temporal Logic Based Learning for Terminal Airspace Operations," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019. https://doi.org/10.2514/6.2019-0682, URL https://arc.aiaa.org/doi/10.2514/6.2019-0682.

[5] Deshmukh, R., Sun, D., and Hwang, I., "Data-Driven Precursor Detection Algorithm for Terminal Airspace Operations," 2019, p. 7.

[6] Murça, M. C. R., "Data-driven modeling of air traffic flows for advanced Air Traffic Management," Thesis, Massachusetts Institute of Technology, 2018. URL https://dspace.mit.edu/handle/1721.1/120378, accepted: 2019-02-14T15:22:40Z.

[7] Murça, M. C. R., and Oliveira, M. d., "A Data-Driven Probabilistic Trajectory Model for Predicting and Simulating Terminal Airspace Operations," *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, IEEE, San Antonio, TX, USA, 2020, pp. 1–7. https://doi.org/10.1109/DASC50938.2020.9256644, URL https://ieeexplore.ieee.org/document/9256644/.

[8] Zhao, W., Li, L., Alam, S., and Wang, Y., "An incremental clustering method for anomaly detection in flight data," *Transportation Research Part C: Emerging Technologies*, Vol. 132, 2021, p. 103406. https://doi.org/10.1016/j.trc.2021.103406, URL https://linkinghub.elsevier.com/retrieve/pii/S0968090X21004010.

[9] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231. https://doi.org/dl.acm.org/doi/10.5555/3001460.3001507.

[10] Rahmah, N., and Sitanggang, I. S., "Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra," *IOP Conference Series: Earth and Environmental Science*, Vol. 31, 2016, p. 012012. https://doi.org/10.1088/1755-1315/31/1/012012, URL https://iopscience.iop.org/article/10.1088/1755-1315/31/1/012012.

[11] Rousseeuw, P. J., "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, Vol. 20, 1987, pp. 53–65. https://doi.org/10.1016/0377-0427(87)90125-7, URL https://linkinghub.elsevier.com/retrieve/pii/0377042787901257.

[12] Sun, J., Ellerbroek, J., and Hoekstra, J., "Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets," *Journal of Aerospace Information Systems*, 2017, pp. 1–6.

[13] Laxhammar, R., and Falkman, G., "Sequential Conformal Anomaly Detection in Trajectories based on Hausdorff Distance," IEEE, Chicago, IL, USA, 2011. URL https://ieeexplore.ieee.org/document/5977571.

[14] Laxhammar, R., and Falkman, G., "Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories," *Annals of Mathematics and Artificial Intelligence*, Vol. 74, No. 1-2, 2015, pp. 67–94. https://doi.org/10.1007/s10472-013-9381-7, URL http://link.springer.com/10.1007/s10472-013-9381-7.

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

[16] Remy, P., "Temporal Convolutional Networks for Keras," https://github.com/philipperemy/keras-tcn, 2020.

[17] Chollet, F., et al., "Keras," , 2015. URL https://keras.io.

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

[19] Boettcher, F., Renner, C., Waldl, H.-P., and Peinke, J., "On the Statistics of Wind Gusts," *Boundary-Layer Meteorology*, Vol. 108, No. 1, 2003, pp. 163–173. https://doi.org/10.1023/A:1023009722736, URL http://link.springer.com/10.1023/A:1023009722736.

[20] Seregina, L. S., Haas, R., Born, K., and Pinto, J. G., "Development of a wind gust model to estimate gust speeds and their return periods," *Tellus A: Dynamic Meteorology and Oceanography*, Vol. 66, No. 1, 2014, p. 22905. https://doi.org/10.3402/tellusa.v66.22905, URL https://a.tellusjournals.se/article/10.3402/tellusa.v66.22905/.