JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 19, No. 2, February 2022

Multiclass Anomaly Detection in Flight Data Using Semi-Supervised Explainable Deep Learning Model

Milad Memarzadeh,* Bryan Matthews,† and Thomas Templin‡
NASA Ames Research Center, Moffett Field, California 94035

https://doi.org/10.2514/1.I010959

The identification of precursors to safety incidents in aviation data is a crucial task, yet extremely challenging. The main approach in practice leverages domain expertise to define expected tolerances in system behavior and flags exceedances from such safety margins. However, this approach is incapable of identifying unknown risks and vulnerabilities. Various machine-learning approaches have been investigated and deployed to identify anomalies, with the great challenge of procuring enough labeled data to achieve reliable and accurate performance. This paper presents an explainable deep semi-supervised model for anomaly detection in aviation, building upon recent advancements described in the machine-learning literature. The proposed model combines feature engineering and classification in feature space, while leveraging all available data (labeled and unlabeled). Our approach is validated with case studies of anomaly detection during the takeoff and landing phases of commercial aircraft. Our model outperforms the state-of-the-art supervised anomaly-detection model, reaching significantly higher accuracy and fewer false alarms, even if only small proportion of data in the training set is labeled.

I. Introduction

THE modern National Airspace System (NAS) is an extremely safe system, and the aviation industry has experienced a steady decrease in accidents over the years. According to the National Transportation Safety Board (NTSB), the accident rate per 100,000 flight hours has been cut in half since 2000, from 0.306 to 0.156 in 2018 [1]. This success trend can be attributed to both greater automation with redundant hardware and software protections as well as to a greater focus on proactive monitoring of and response to real-time and historically identified vulnerabilities. Although the number of passenger enplanements has increased by 20% from 706 million in 2009 to 851 million in 2017, the number of departures has decreased by 5% from 9.7 million to 9.3 million over the same period [2]. This development has resulted in a historically high passenger load factor (82.3% in 2017) [3]. As the load factor approaches saturation, one would expect that the number of departures will rise again in the near future. To maintain its historically low level of accidents per year, the NAS will need to continually evolve. This includes proactively identifying previously unknown and operationally significant safety events occurring during daily flight operations.

Identifying situations where unknown risks or vulnerabilities exist is not a trivial problem. Much of the knowledge about adverse events comes from after-the-fact forensic analysis aimed at determining the root causes of incidents or accidents. The NTSB uses this approach when investigating an accident [4]. In this area, machine learning can improve the process of risk identification and eventually help with risk mitigation by identifying emerging vulnerabilities and safety requirements that need revising. However, automatically identified risks still need to be reviewed and assessed by subject matter experts familiar with aircraft behavior and NAS operations. This step is needed to better understand how operations are being carried out and the safety implications associated with the status quo. New vulnerabilities can then be addressed by mitigating the factors identified to contribute to safety events with proper countermeasures. Remediation includes improving pilot/controller training and developing automation safety procedures. If such measures are put in place, they can help avoid unsafe flight states that bring about an increased likelihood of an incident or accident, which in turn may result in damage to the aircraft, injury, or loss of life.

One of the main challenges in using machine learning to identify precursors to safety events in the aviation domain is the sparse quantity of processed and labeled data, for which anomalous patterns have been reliably identified and labeled. As a result, the pertinent literature primarily focuses on unsupervised reasoning to identify anomalies in high-dimensional time series of flights. Unfortunately, the high degree of uncertainty inherent in unsupervised learning results in a high number of false alarms and low accuracy in complex settings, which limits their applicability. On the other hand, supervised reasoning cannot reach typically observed peak performance due to the lack of labeled data. To address the shortcomings of both unsupervised and supervised reasoning when confronted with aviation data, we have developed a robust and explainable semi-supervised model. This model takes advantage of the entirety of available aviation time-series data, including a majority of unlabeled data and a minority of labeled data, to identify anomalies or incident precursors with high accuracy and few false alarms.

In the aviation safety domain, it is important to address model explainability to build trust and acceptance within the community. If model transparency is inadequate and automated decision-making murky, industry will be reluctant to adopt new technologies even if performance metrics are strong. Furthermore, understanding patterns and interactions of signals and their propagation from the input-parameter space aids in the validation of anomalies and assists with designing mitigation strategies that proactively prevent adverse events from occurring in the future. We employ two approaches to make our models more explainable: we 1) quantify the contributions to known adverse events from the input parameter space, and 2) visualize the configuration of the latent feature space to better understand how the algorithm organizes the data.

In the remainder of the paper, we first review the literature on aviation anomaly detection. Next, we discuss our semi-supervised models, present their performance on binary and multiclass anomaly-detection problems using flight operational quality assurance (FOQA) data, and compare their performance to the state-of-the-art supervised aviation anomaly-detection model. Lastly, we discuss the explainability of these models and identify next steps for further investigations.

Presented as Paper 2021-0774 at the AIAA SciTech 2021 Forum, Virtual Event, January 11–15 and 19–21, 2021; received 15 January 2021; revision received 2 June 2021; accepted for publication 22 August 2021; published online Open Access 28 September 2021.

*Senior Scientist, Data Sciences Group, Universities Space Research Association; milad.memarzadeh@nasa.gov (Corresponding Author).

†Research Engineer, Data Sciences Group, KBRwyle.

‡Research Computer Scientist, Data Sciences Group. Senior Member AIAA.

II. Related Work

The established approach for identifying vulnerabilities in operations leverages domain expertise about how the system should behave within known tolerances and safety margins. This approach is known as exceedance detection, which is the standard technique for anomaly detection in aerospace data [5]. This technique compares parameters relevant to flight with predefined thresholds, which are identified based on domain knowledge. The exceedance-detection method works well on known issues and when the system has a well-defined operating condition but is incapable of identifying unknown risks and vulnerabilities.

To identify unknown hazards, we need to go beyond simplistic approaches. Recent advances in the field of machine learning suggest their application for identifying anomalies in aviation data. In general, machine learning approaches to aviation anomaly detection can be categorized into supervised and unsupervised methods, with the presence or absence of labeled data the key differentiator between the two. Data that are reviewed and annotated by subject matter experts who identify and time-stamp anomalies—or certify their absence—are considered labeled data. However, the huge volume of recorded aviation data renders widespread review and labeling impossible, and data that lack information on the presence or absence of anomalies are considered unlabeled data.

Supervised-learning models produce inference using only labeled data and have demonstrated impressive performance when trained on a sufficiently large number of data points. Lee et al. [6] developed several classic supervised-learning approaches to identify safety anomalies in FOQA data. Janakiraman [7] developed a deep temporal multiple-instance learning (DT-MIL) approach by exploiting a recurrent neural network for the supervised classification of adverse events in FOQA data. Mori [8] proposes a recurrent neural network architecture to estimate a lateral stability index to identify unstable approaches during landing. The developed method uses feature engineering to specifically identify unstable approaches and might not generalize to other types of anomalies during landing and/or other phases of flight. These supervised models performed well in identifying safety-incident precursors (anomalies) in aviation time series. However, labeling aviation data requires time-consuming and costly efforts of subject matter experts and is largely impractical. As a result, the size of reliably labeled aviation datasets is not large enough to allow supervised models to reach optimum performance.

Because of the difficulty of obtaining labels for aviation data, unsupervised-learning approaches for anomaly detection have been another thrust of research. The field of unsupervised approaches is diverse and includes proximity-based methods [9,10], clustering-based methods [11,12], kernel-based methods [13–15], and deep-learning-based methods [16,17]. Bay and Schwabacher [9] define anomaly as a point in a feature space with remote nearest neighbors. Their model was specifically designed to detect anomalies is Space Shuttle engines. Melnyk et al. [10] developed an autoregressive model for multivariate time series of flights and used a distance metric to identify dissimilar flights using a nearest-neighbor approach. Li et al. [18] developed the Cluster-AD-Flight method that transforms FOQA data into high-dimensional vectors, making different flights comparable by sampling each flight parameter at fixed temporal or distance-based intervals starting from an anchoring event (e.g., time from takeoff or distance from touchdown) with subsequent clustering using a density-based spatial clustering algorithm. Kernel-based methods based on support vector regression [15] and one-class support vector machines (OC-SVMs) [13,14] have also been developed to identify anomalies in FOQA data. An OC-SVM constructs an optimal hyperplane that segregates normal data in a high-dimensional reproducing kernel Hilbert space by maximizing the margin between the origin and the hyperplane.

These approaches use different unsupervised techniques to identify the majority nominal pattern in the unlabeled pool of data and then use distance- or proximity-based metrics to identify data points that are far away from the majority pattern. Although these models address the problem of the paucity of labeled data, they suffer from a high number of false alarms and low accuracy of anomaly detection, especially when the anomaly is not a point anomaly (anomaly that occurs during one time-stamp). This is especially evident when confronted with complex data such as high-dimensional heterogeneous time series. This drawback is typically due to the fact that statistically significant anomalies are not guaranteed to coincide with operationally significant anomalies. For example, in a previous study we showed that the performance of our unsupervised model (a variational autoencoder with convolutional architecture) increased by ∼32% when the model was only trained on the majority of nominal data [17]. This finding illustrates that incorporation of weak supervision (removing anomalies from training) had a significant impact on the model's performance.

Motivated by the above finding, we have developed semisupervised models that use labeled nominal data, labeled anomalous data, and unlabeled data for learning. Semi-supervised models usually combine unsupervised feature engineering (based on all available data) with supervised classification (based on labeled data only). Figure 1 shows the overall design of the semi-supervised models proposed here. The main hypothesis is that by leveraging a large pool of unlabeled data, unsupervised feature engineering can be incorporated into the model to complement supervised classification. Coupling supervised classification with unsupervised feature engineering in feature space can propel the model to reach optimal performance, given the scarcity of labeled data. Specifically, we build upon an unsupervised feature-engineering model, convolutional variational auto-encoder (CVAE), previously developed by the authors [17], as well as upon two successful semi-supervised classification models described in the deep-learning literature, named M1+M2 [19] and Compact Clustering via Label Propagation (CCLP) [20], for detecting anomalies in high-dimensional and heterogeneous time series.

To benchmark the performance of these models, we have developed two case studies: 1) binary anomaly detection during the takeoff phase of commercial aircraft, and 2) multiclass anomaly detection during approach to landing of commercial aircraft. We compare the performance of our semi-supervised models with the performance of the state-of-the-art supervised model [7], developed specifically for anomaly detection in aviation data.

III. Method

In this section, we summarize two semi-supervised models that we adopted from the deep-learning literature for the purpose of anomaly detection. The assumption is that the available data are grouped into two sets: labeled data, (X_L, y_L), and unlabeled data, X_U, where the size of the unlabeled set is significantly larger, $N_U = |X_U| \gg N_L = |X_L|$. As discussed above, unsupervised learning ignores y_L, whereas supervised learning ignores X_U. The goal of our semi-supervised models is to not only use both sets of data, but to make sure that deployment of both X_U and y_L improves performance compared with unsupervised and supervised models [21]. In the next sections, we summarize two models: M1+M2 [19] and CCLP [20]. The M1+M2 model is a deep generative model built upon recent successes in deploying variational auto-encoders (VAEs) [22]. The CCLP model is a deep encoding model that enforces compact clustering of data belonging to the same class in the latent feature space. As illustrated in Fig. 1, both of these models contain two main components: 1) a feature encoder that encodes the high-dimensional input data into a lower-dimensional latent feature space, and 2) a classifier applied in the latent feature space that identifies whether a data point is anomalous or not and—in the case of multiclass classification—predicts anomaly type. We take advantage of our previously developed CVAE model architecture [17] for feature encoding in both of these semi-supervised models. The main difference between the two models is in the objective functions used for training the network parameters, which we will describe in detail in the next subsections.

We compare the performance of these two models with the state-of-the-art supervised learning model, DT-MIL [7]. DT-MIL uses a deep recurrent neural network to take temporal dependencies of the data into account and was specifically developed for anomaly detection in aviation data. We benchmark the performance of these models

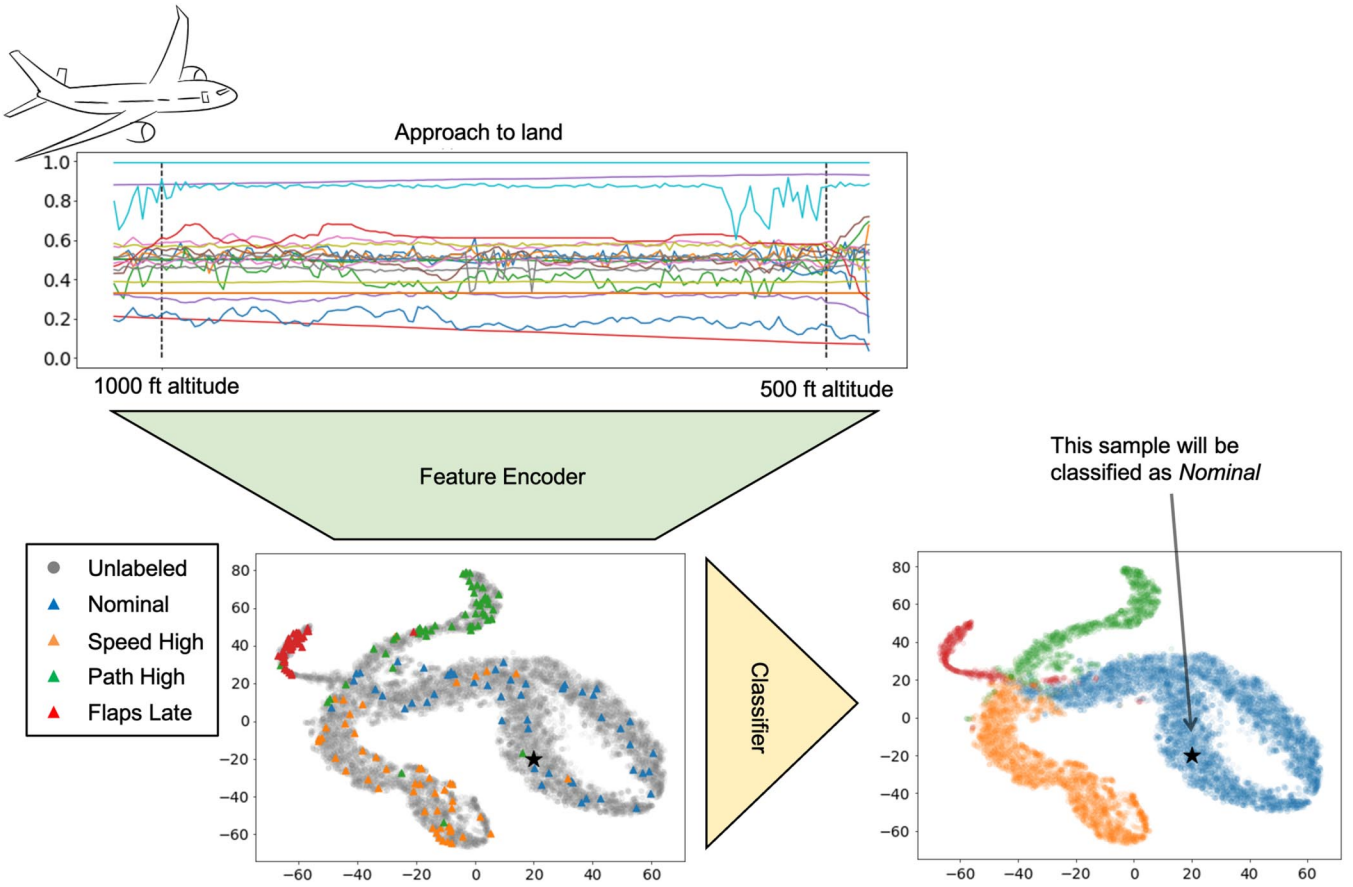


Fig. 1 Graphical illustration of the proposed semi-supervised anomaly-detection models. The latent-space configurations in this figure are from the approach using the CCLP model.

using two sets of FOQA data. The first dataset comprises nominal data points (time series) as well as one type of anomaly during the takeoff phase of commercial aircraft. The second dataset comprises nominal data points (time series) as well as multiple types of anomalies during the approach to landing of commercial aircraft. We will discuss details of the data later on.

A. M1+M2: Semi-supervised Deep Generative Model

As evidenced by its name, M1+M2 [19] is a combination of both an M1 and an M2 model. M1 is a β-VAE [23], serves mainly as an unsupervised feature encoder, and maps the input data to a lower-dimensional latent space. The M2 model is a VAE deployed in the latent space of the M1 model, plus a classifier. Figure 2A illustrates

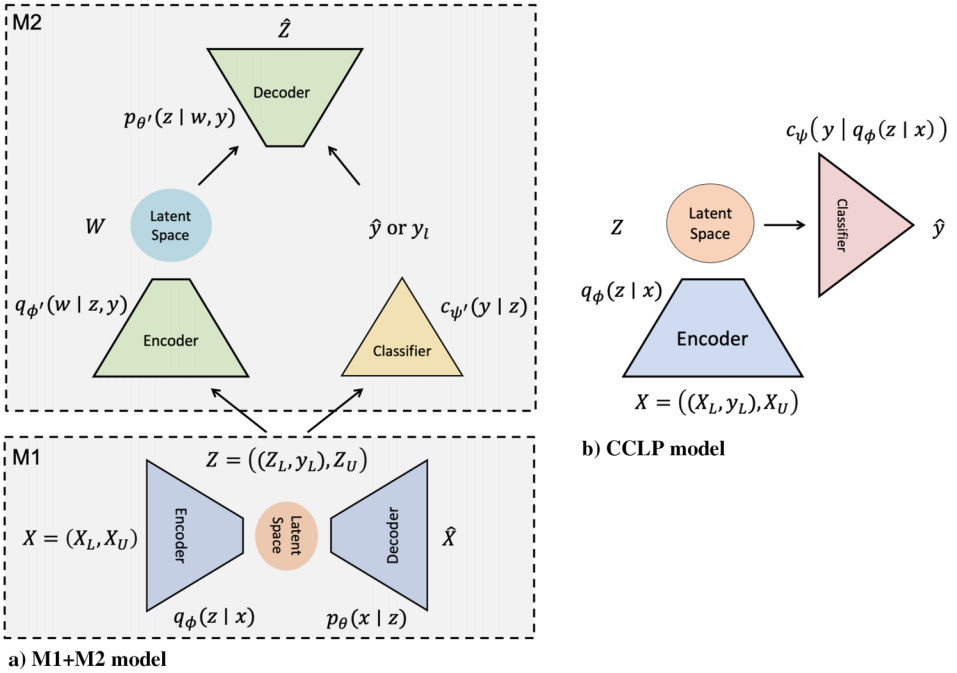


Fig. 2 Graphical illustration of A) M1+M2 and B) CCLP model architectures.

the M1+M2 model graphically. The two submodels (M1, M2) are trained separately.

The M1 model is trained in an unsupervised fashion and serves the role of feature engineering. It consists of an encoder that maps the input data X to a lower-dimensional latent feature space Z and a decoder that reconstructs the input data $\hat{X}$ by sampling from the latent space. It is trained based on maximizing the following objective function:

$$\mathcal{J}_{\mathrm{M1}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta \mathrm{KL}(q_\phi(z|x) \| p_\theta(z)) \tag{1}$$

The first term is the expected value with respect to the encoder distribution of the log likelihood of the reconstructed data conditioned on the samples from the latent space (i.e., the expected reconstruction fidelity). The second term is the KL divergence between the posterior distribution of the latent space $[q_\phi(z|x)]$ and the prior distribution of the latent space $[p_\theta(z)]$, which is assumed to be a standard normal distribution. The factor β is a hyperparameter controlling the importance of the KL-divergence term in the above objective function; $q_\phi(z|x)$ and $p_\theta(x|z)$ could be any functions that serve as encoder and decoder, respectively. However, in this paper, we use a deep neural network to represent these functions, with ϕ and θ as the parameters (or weights) of the neural networks.

The M2 model is a VAE plus a classifier and built in the latent space of the M1 model. The objective function of the M2 model is depicted as follows:

$$\mathcal{J}_{\mathrm{M2}} = \sum_{(z_l, y_l)} \mathcal{L}(z_l, y_l) - \alpha \mathbb{E}_{(z_l, y_l)}[\mathcal{H}(y_l, c_{\psi'}(y|z_l))] + \sum_{(z_u)} \mathcal{U}(z_u) \tag{2}$$

The first term is a VAE objective function for the labeled set as defined in Eq. (1) with $\beta = 1$ and slight variations to incorporate labels:

$$\mathcal{L}(z, y) = \mathbb{E}_{q_{\phi'}(w|z,y)}[\log p_{\theta'}(z|w, y)] - \mathrm{KL}(q_{\phi'}(w|z, y) \| p_{\theta'}(w) p_{\theta'}(y)) \tag{3}$$

It should be noted that the latent space variables of the M2 model, i.e., w, and the classifier's prediction, i.e., $\hat{y}$, are concatenated together and then passed on as one single input to the decoder. The second term is the classification loss, which is defined as the expected value of the cross entropy between the actual data labels (y_l) and the class probability scores $(c_{\psi'}(y|z_l))$. The term ψ' denotes the parameters (or weights) of the classifier, and α is a hyperparameter that denotes reliance on discriminative learning (operationalized by the classification loss). The third term is the VAE objective function for the unlabeled set and is defined as follows:

$$\mathcal{U}(z) = \sum_y c_{\psi'}(y|z)\mathcal{L}(z, y) - \mathcal{H}(c_{\psi'}(y|z)) \tag{4}$$

where the first term marginalizes the VAE objective for the labeled set over all possible labels and the second term (entropy of the labels predicted by the [discriminative] classifier) weights predictions for the unlabeled data proportionally to the confidence with which they were made, to improve reconstruction.

B. CCLP: Semi-supervised Model Through Compact Latent-Space Clustering

The CCLP approach [20] follows the cluster assumption in semi-supervised machine learning, which in Sec. 1.2.2 of [21] states that "if the data of the same class tend to form a cluster in the latent space, then the unlabeled set could aid in finding the boundary of each cluster more accurately." Figure 2B illustrates the architecture of the CCLP model; it consists of an encoder that maps the input data to a lower-dimensional latent feature space, and a classifier that classifies the data in the latent space.

The key idea behind this approach is to dynamically create a graph over the embedding of labeled and unlabeled samples to capture the underlying structure in the latent feature space and to use label propagation to estimate its high- and low-density regions. This idea is pursued by the addition of an extra term in the objective function, which enforces the compact clustering of the data of the same class in the latent space. This is given by the following equation:

$$\mathcal{J}_{\mathrm{CCLP}} = -\mathbb{E}_{(x_l, y_l)}[\mathcal{H}(y_l, c_\psi(y|q_\phi(z|x_l)))] - \eta \mathcal{L}_{\mathrm{CCLP}} \tag{5}$$

where the first term is the classification loss (cross entropy) for the labeled set and the second term is the cross entropy between the optimal transition function T and the one estimated via label propagation H. Hyperparameter η tunes the importance of this term in the whole objective function. The optimal transition function T denotes a desired optimal state, where a single compact cluster is formed per class in the latent feature space. In such an optimal state, the transition probability between any two data points of the same class is the same, and it is zero for inter-class transitions. The cross entropy is specified as follows:

$$\mathcal{L}_{\mathrm{CCLP}} = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} -T_{ij} \log H_{ij}^{(s)} \tag{6}$$

where $N = N_L + N_U$ is the total number of samples, T the optimal transition matrix, H the actual transition matrix (estimated via dynamic graph construction and label propagation), and s the step of the Markov chain on the graph. Matrix H is calculated by normalizing adjacency matrix A, which is estimated based on the similarity of the data points in the latent space. In this paper, we have used the cosine similarity as a similarity metric, but other metrics such as negative Euclidean distance can be used as well. On the other hand, the calculation of T is based on the propagation of the class posteriors from the labeled set to the unlabeled set according to random walks formed by the estimated transition matrix H, until the equilibrium state is obtained. Fortunately, class posteriors for the unlabeled set at equilibrium can be calculated in closed-form without the need for computationally intensive iterations. For further details regarding the computation of the estimated and optimal transition functions H and T, refer to [20].

IV. Results and Discussion

In this section, we first note the details of the semi-supervised models (M1+M2 and CCLP) that are implemented here. Next, we introduce two sets of data for anomaly detection, one for the takeoff phase and one for the approach to landing of commercial aircraft, evaluate the performance of the models, and compare it to the state-of-the-art supervised model (DT-MIL) on both problems. Then, we discuss the latent feature space configurations of these models and how they can be interpreted for postprocessing.

A. Implementation Details

Both semi-supervised models introduced here were implemented in Python using the PyTorch library (https://pytorch.org/). On the other hand, the supervised DT-MIL model was used the way it was originally implemented, using the Keras library (https://keras.io/). Although the semi-supervised and supervised models were implemented using different libraries, this difference does not affect their performance. All models were trained for 200 epochs, and we monitored the training loss to confirm the convergence of training. The Adam optimizer [24] was used for all models listed in the following subsections.

1. M1+M2 Model

As noted before, M1+M2 consists of two separate (sub)models. The M1 model performs unsupervised feature engineering, mapping the input data to a latent feature space. Although the M2 model serves primarily as a classifier (nominal or anomalous time series, or what type of anomaly in the case of multiclass anomaly detection), it

should be noted that the M2 model is superior to simply adding a classifier in the latent space of the M1 model (as shown by the original paper [19]). This is because the additional model adds extra terms in the objective function, therefore forcing the classifier to make confident predictions, which aids in the construction of the latent feature space and reduces the reconstruction error.

We have used our previously developed CVAE model architecture [17] to serve as the M1 model in this study. The details of the model's architecture are illustrated in Fig. A1 in the Appendix. The hyperparameter β in Eq. (1) was set to 0.001, based on our comprehensive experimentation with β-VAE in [17]. Usually a large value of β (close to 1) results in superior data generation and sampling from the latent space posterior, whereas smaller values result in more compact clustering of data in the latent space and improve classification performance. We used the binary cross entropy to estimate the expected (log-) likelihood of reconstruction [i.e., the first term in Eq. (1)]. The latent space dimension of the M1 model was fixed to 128 ($z \in \mathbb{R}^{128}$) for the binary problem and to 256 ($z \in \mathbb{R}^{256}$) for the multiclass problem. The reason that the dimensionality of the latent space in the multiclass problem is higher than in the binary problem is that the dimension of the multiclass problem's input data is much higher than the dimension of the binary problem's input (we will discuss these datasets in the next section). Moreover, our hyperparameter tuning effort exhibited the optimal performance with these dimensions.

The M2 model was implemented using a fully connected architecture. The encoder consists of two fully connected layers, each with 100 neurons and ReLU activations. The encoder's second fully connected layer connects to a linear layer, with the number of neurons equal to the dimensionality of the latent space. The decoder is identical to the encoder, but the layers' order is reversed. The classifier has two fully connected layers, each with 100 neurons and ReLU activations, with a dropout layer (50%) between them. The second fully connected layer then connects to the output layer with either one neuron with sigmoid activation (binary problem) or with multiple neurons with softmax activation, whose number equals the number of classes (multiclass problem). The hyperparameter α in Eq. (2) was fixed to $0.1 \times N_U$ for all experiments (as suggested by original developers of the model), where N_U is the size of the unlabeled set.

2. CCLP Model

The CCLP model consists of two parts: 1) an encoder identical to the M1-model encoder (illustrated in Fig. A1 in the Appendix) and 2) a classifier identical to the M2-model classifier described in the previous subsection. Hyperparameter η in Eq. (5) was fixed to 1, and the number of steps in the transition function, S in Eq. (6), was fixed to 3, both corresponding to the original developers' recommendations and experimentation [20].

3. DT-MIL Model

The architecture of the DT-MIL model remains unchanged from the model's original architecture (as conceived by its developer [7]). It consists of a GRU layer with five neurons and tanh activations, followed by a time-distributed fully connected layer of 500 neurons with tanh activations, connected to a time-distributed fully connected layer with one neuron and sigmoid activation, which calculates the precursor scores for all time steps in a time series. Then the result of the precursor layer is pushed through a global max-pooling layer to classify the data.

It should be noted that DT-MIL is a binary classification model. Because we intended not to alter the original model, we implement DT-MIL as a one-vs-all scheme for the case of multiclass classification (time series of approach to landing of commercial aircraft).

B. Binary Anomaly Detection During Takeoff

In this problem, we focus on identifying one type of anomaly during takeoff of a commercial aircraft. The anomaly was identified by subject matter experts and defined as a threshold on the drop in airspeed during the first 60 s of takeoff: A drop in airspeed by more than 20 knots during the first 60 s of takeoff could lead to an undesirable incident. Based on this insight, we prepared a binary anomaly-detection dataset to test how accurately different models can identify this specific type of anomaly during the takeoff phase. This is not necessarily the only type of anomaly in the dataset, and the true number of operationally significant safety anomalies is unknown; however, our objective was to measure the effectiveness of the algorithms described above, and we used this particular safety incident as one measure of the algorithm's performance.

Accordingly, we preprocessed FOQA data to set up training/testing datasets for benchmarking the performance of semi-supervised models (M1+M2 and CCLP) relative to the performance of the state-of-the-art supervised model (DT-MIL). Each data sample is a 60-s-long multivariate time series of 19 variables measuring roll attitude (roll angle), altitude, pitch attitude (corrected angle of attack [AoA] and pitch angle), speed information (computed airspeed), low rotor speed (N1 actual) and high rotor speed (N2 actual), yaw attitude (drift angle, rudder position, and wind speed), control-surface variables (right and left ailerons positions, right and left elevator positions, and stabilizer position), and flap configurations (deployed at 10 deg [Flaps0], 15 deg [Flaps1], 20 deg [Flaps2], and 40 deg [FlapsFull]). It should be noted that FOQA data contain hundreds of variables, and the 19 variables used in our study were selected due to their relevance to the takeoff phase of flight based on the guidance of subject matter experts.

The training data consist of 16,407 samples, among which only 402 are anomalous (roughly 2.5%). Similarly, the test data consist of 5470 samples, among which 126 are anomalous. The training data are used for training each model, and the test data are used to estimate the unbiased performance of the models. All performance metrics shown in the figures are based on the performance of the models on the test data. To evaluate the models' anomaly-detection performance, we assume that the nominal data are the *negative* class and the anomalous data are the *positive* class. As a result, two important metrics can be calculated as follows:

$$\text{precision} = \frac{TP}{TP + FP}$$
$$\text{recall} = \frac{TP}{TP + FN} \tag{7}$$

where TP is the number of true positives (anomalies that are correctly identified), FP the number of false positives (anomalies that are incorrectly identified or false alarms), and FN the number of false negatives (anomalies that are missed and classified as nominal by mistake). Higher precision means that the model generates a lower number of false positives, whereas higher recall means that the model misses fewer anomalies.

Although we have labels available for the entire training dataset, for the sake of experimentation, we evaluated the performance of the models when only a small portion of the entire training set was labeled. To this end, we divided the training set into sets of labeled data, (X_L, y_L), and unlabeled data, X_U, where $N_L + N_U = N =$ 16,407. Figure 3 compares the performance of the three models in terms of precision (top panel) and recall (bottom panel) when the size of the labeled set, N_L, varies between 100 (0.6% of the training data) and 1000 (6% of the training data). The labeled set was randomly sampled from the pool of training data, and, to obtain reliable statistics, we repeated the training process for each model 50 times. The bar charts in Fig. 3 show the mean $\pm$ the standard deviation of the models' performance on the test data, based on these 50 independent experiments. We also visualize the precision–recall curves and the area under the curve in Fig. A2 in the Appendix.

As illustrated by the figure, both semi-supervised models outperform the supervised model in both precision and recall, and M1+M2 performs consistently better than CCLP in this problem. It is quite interesting to see that, on average, M1+M2 reaches nearly 97.5% precision and 70% recall with only 500 labeled data (3% of the total training set). It should be noted that among these 500 labeled data, there are only 2.5% or ~12 anomalous examples, because the pool of labeled data is selected randomly. This result demonstrates the incredible performance of the M1+M2 model in this case study; it is able to identify anomalies with high precision and recall with a

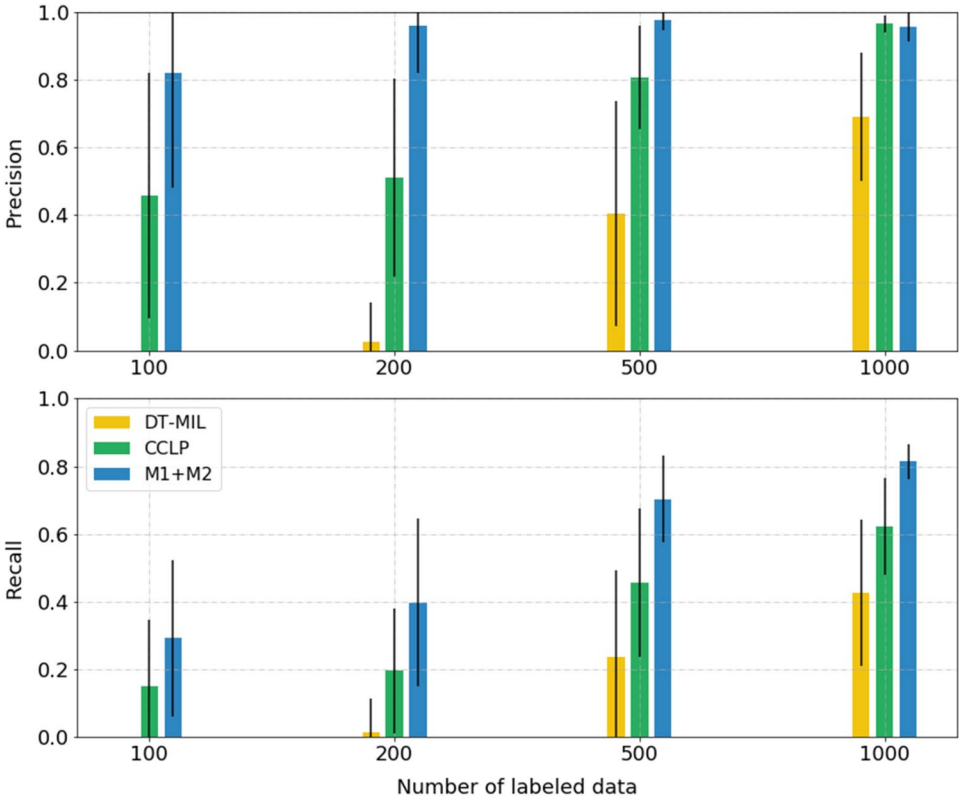


Fig. 3 Performance of the semi-supervised models ($M1 + M2$ and CCLP) and of the supervised model (DT-MIL) on the binary anomaly-detection problem.

minimal number of labeled data. We can also infer that, as the number of labeled data increases, the performance of the CCLP model gets closer to the performance of the M1+M2 model, whereas the supervised model is significantly less accurate compared with the two semi-supervised models, and it performs very poorly in the case of only 100 or 200 labeled time series.

Figure 4 shows the importance of the input variables for anomaly-detection precision and recall. We used the random-permutation method to obtain this figure. In this method, each attribute of the input data is randomly and independently permuted across the entire dataset; then we use the trained encoder to encode the input attributes into latent-space features; finally, a measurement is made on by how much the classification precision and recall values drop in response to this random permutation by evaluating the classifier's performance on the latent-space features. If an attribute is not important, the drop in classification performance will be close to zero, while important attributes exhibit greater drops in performance. According to this method, the most important features that help identify anomalies are the computed airspeed, pitch angle, and wind speed. This result is not unexpected because the anomaly under consideration is based on a drop in airspeed, and all these three factors record information relevant to speed of the aircraft. It should be noted that while high precision indicates the correct classification of data belonging to the majority nominal class, high recall signifies the correct classification of data in the minority anomalous class. As a result, recall is more affected than precision when the important features are permuted. As a point of comparison and to better understand how these features help with classifying anomalies, Fig. A3 in the Appendix illustrates the mean $\pm$ the standard deviation of the trajectories of these top three variables for nominal versus anomalous data examples in the test data.

C. Multiclass Anomaly Detection During Approach to Landing

In this section we introduce a multiclass anomaly-detection dataset based on FOQA data from a commercial airline (https://c3.nasa.gov/dashlink/projects/85/). These data comprise primarily 1 Hz recordings for each flight (similar to the dataset for binary anomaly detection) and cover a variety of systems. These include the state and orientation of the aircraft, positions and inputs of the control surfaces, engine parameters, and autopilot modes and corresponding states. The data are acquired in real time on-board the aircraft and downloaded by the airline once the aircraft has reached the destination gate. These time series are analyzed by domain experts, which derive threshold-based rules postflight to flag known events and create labels. Each data instance is a 160-s-long recording of 19 variables during the approach of the aircraft to landing, from a few seconds before an altitude of 1000 ft to a few seconds after an altitude of 500 ft. It should be noted that for many flights the duration from 1000 to 500 ft altitude is less than 160 s; in that case we expand the data window to include an additional time period directly before reaching 1000 ft altitude.

Using the threshold-based rules obtained from subject matter experts, we created multiclass anomaly detection training and testing datasets containing 18,313 and 6105 samples, respectively. These data comprise four classes: 1) nominal, where no anomaly of the other three classes is known to be present (~66.7% of the total data); 2) speed high, where the anomaly is identified based on computed airspeed being above a threshold during approach (~22.9% of the total data); 3) path high, where the glide slope and path of descent for landing are flagged as being high (~7.2% of the total data); and 4) flaps late, where the flaps are deployed late during approach to landing (~3.2% of the total data). These events were chosen because they are all relevant metrics used to measure unstabilized approaches. Each data instance is either nominal or contains only one type of anomaly, a restriction that simplifies the validation process; testing on data that contain multiple types of anomalies per instance will be part of our future work. Figure 1 shows a sample of normalized trajectories from the nominal class. Similar to the binary example, the training data are used to train each model, and the test data are used to estimate the unbiased performance of a model. All results shown in the figures are based on the performance of the models on the test data.

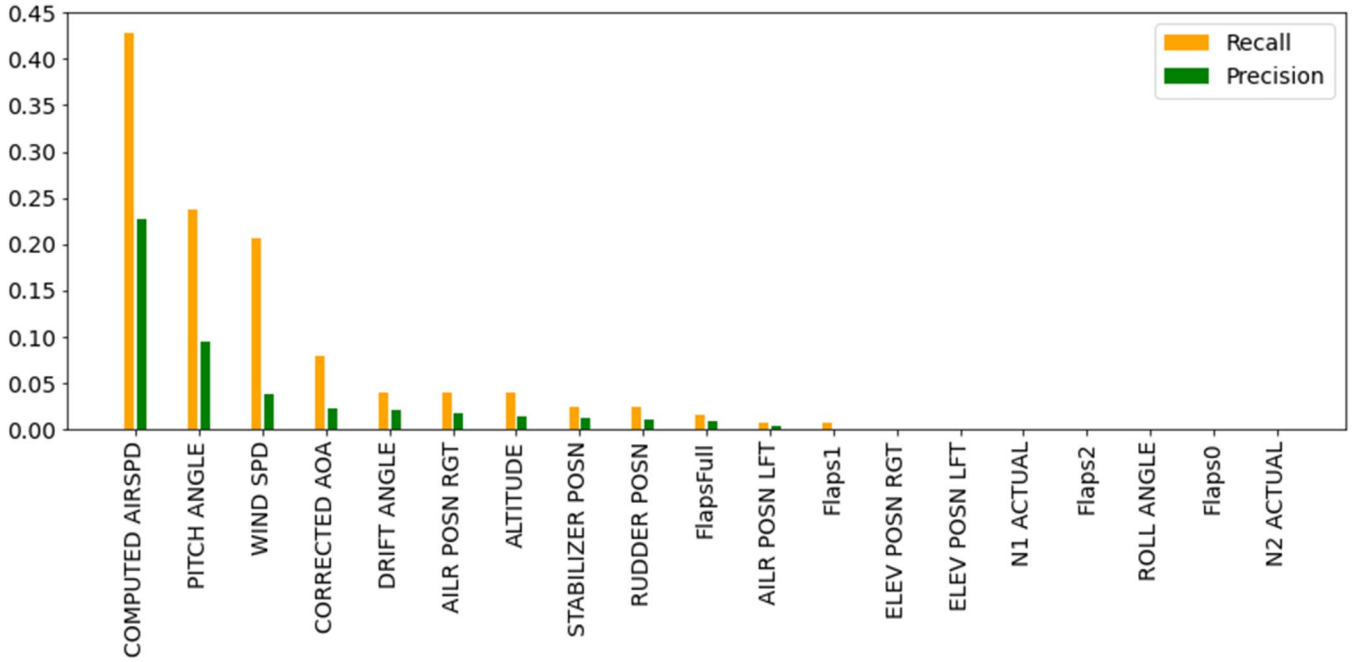


Fig. 4 Importance of parameters for drop-in-airspeed anomaly, as identified by the random permutation method.

As mentioned above, the currently labeled set of the data only contains threshold-based anomalies that, if all known in advance, can be detected by a simple method (e.g., exceedance detection). However, operations in the national airspace are very complex, and as a result, many complex anomalies might occur that are characterized by nonlinear correlation between multiple factors. In such a setting, a simple exceedance-detection model will fail to adapt to detection of such anomalies. Our goal is to develop a general model capable of identifying both simple and complex anomalies in aviation data. We hope that with improved industry engagement, we can improve the quality of Artificial Intelligence-ready data to include more diverse and more complex anomalies.

Although we have labels available for the entire training dataset, for the sake of experimentation, we evaluated the performance of the models when only a small portion of the entire training set was labeled. To this end, we divided the training set into sets of labeled data, (X_L, y_L), and unlabeled data, X_U, where $N_L + N_U = N = 18,313$. Figure 5 compares the performance of the three models' accuracy when the size of the labeled set, N_L, varies between 100 (0.55% of the training data) and 1000 (5.5% of the training data). We also report the precision and recall values per class for all models in Fig. A4 in the Appendix. In this experiment, the labeled set was uniformly sampled across the four classes from the pool of training data, meaning that when the size of the labeled set was 100, there were 25 labeled data points (time series) per class. The data in each class were still selected randomly and, to obtain reliable statistics, we repeated the training process for each model 20 times. The columns in Figure 5 show the mean $\pm$ the standard deviation of the models' performance on the test data, based on these 20 independent experiments.

As can be seen from both Figs. 5 and A4, both semi-supervised models outperform the supervised model (DT-MIL) significantly and are able to achieve high accuracy (CCLP: 72.2%; M1+M2: 68.8%) with only 100 labeled data points (25 per class), which corresponds to 0.55% of the total data, whereas DT-MIL performs poorly (accuracy of 31.4%). To better demonstrate the performance of the two semi-supervised models (M1+M2, CCLP) and their superiority to the supervised DT-MIL model, we show normalized confusion matrices for the case of 1000 labeled data points (250 per class) (Fig. 6). A normalized confusion matrix visualizes the proportions of the data

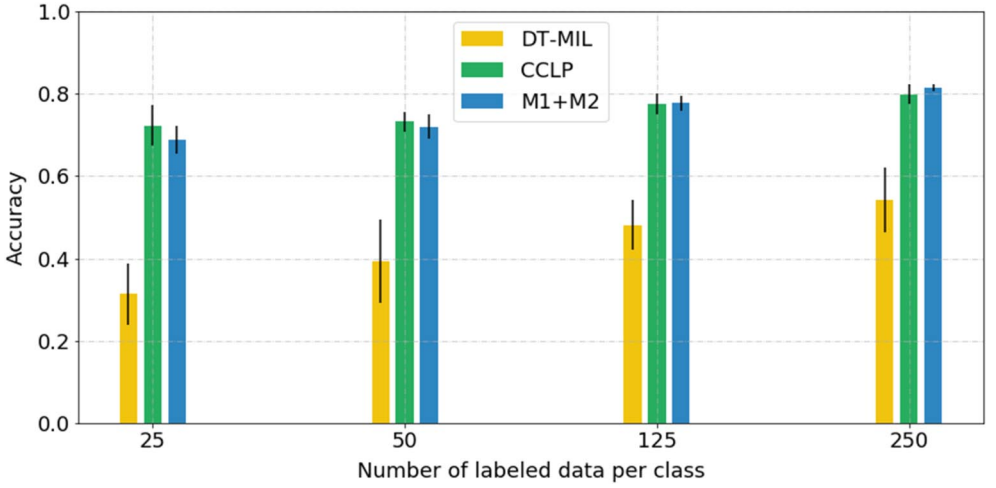


Fig. 5 Performance of the semi-supervised models (M1+M2 and CCLP) and of the supervised model (DT-MIL) on the multiclass anomaly-detection problem.

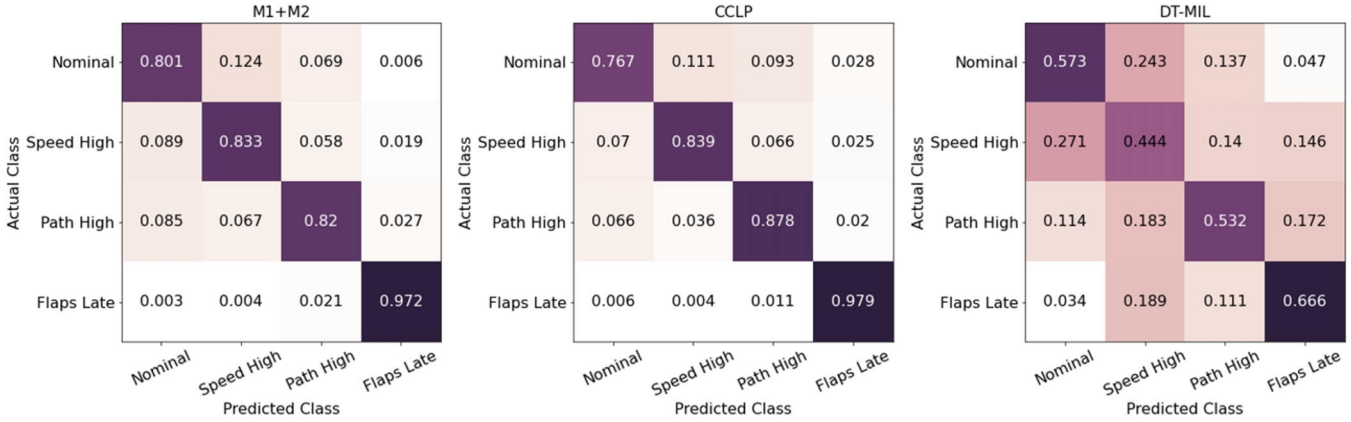


Fig. 6 Confusion matrices for the M1+M2, CCLP, and DT-MIL models for the case of 1000 labeled data points.

that are correctly classified in each class; the *x* axis presents the class predicted by the model, and the *y* axis presents the actual class. The values that lie on the diagonal represent correct classifications and the off-diagonal values errors ("confusions") of the model. The two matrices reveal that M1+M2 exhibits a better performance in identifying the nominal class, whereas CCLP performs better in identifying and distinguishing the anomalous classes, while sacrificing accuracy in detecting nominal examples. Both semi-supervised models significantly outperform DT-MIL.

To quantify parameters important in identifying each of the four classes, we used random permutations (similar to what we did in the case of the binary problem), but combined precision and recall into their harmonic mean, i.e., the F1 score. For this purpose, we first calculated the precision and recall per class, where the class of interest was considered the *positive* class and all other three classes were considered *negative*. Once the precision and recall for each class is calculated according to Eq. (7), the F1 score can be computed as follows:

$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8}$$

Figure 7 shows the importance of the features in terms of F1 score for classifying each of the four classes (nominal plus three anomalous classes) using the random permutation method:

1) *Speed high:* The top three features to identify this class are corrected AoA, aircraft heading (selected and true headings), and computed airspeed, which correlates with intuitive assumptions.

2) *Path high:* The top three features in this class are glide slope deviation (which is the main feature used to flag these events), average core speed (which is the average thrust of the four engines), and pitch angle (another direct indicator of the aircraft deviating from glide slope).

3) *Flaps late:* The top three features are corrected AoA, computed airspeed (as expected because the deployment of flaps increases lift and allows flight at a lower AoA and at a lower airspeed), and trailing edge (T.E.) flap position (which is the main feature used to flag these events).

Figures A5–A7 in the Appendix show the mean ± the standard deviation of the top features in each class, compared with the nominal class. For example, in the case of the flaps late class, one can see that the late deployment of flaps (which is the anomaly) correlates with a higher than normal airspeed in the approach to landing. This is the reason why features related to speed information appear also important in identifying this type of anomaly. It would be interesting to see how much overlap exists between this anomaly and speed high events during approach; the relationship between the two situations could be investigated further in the future.

D. Latent Space Configuration

In this section, we investigate the latent space configurations of the semi-supervised models and use the multiclass anomaly-detection

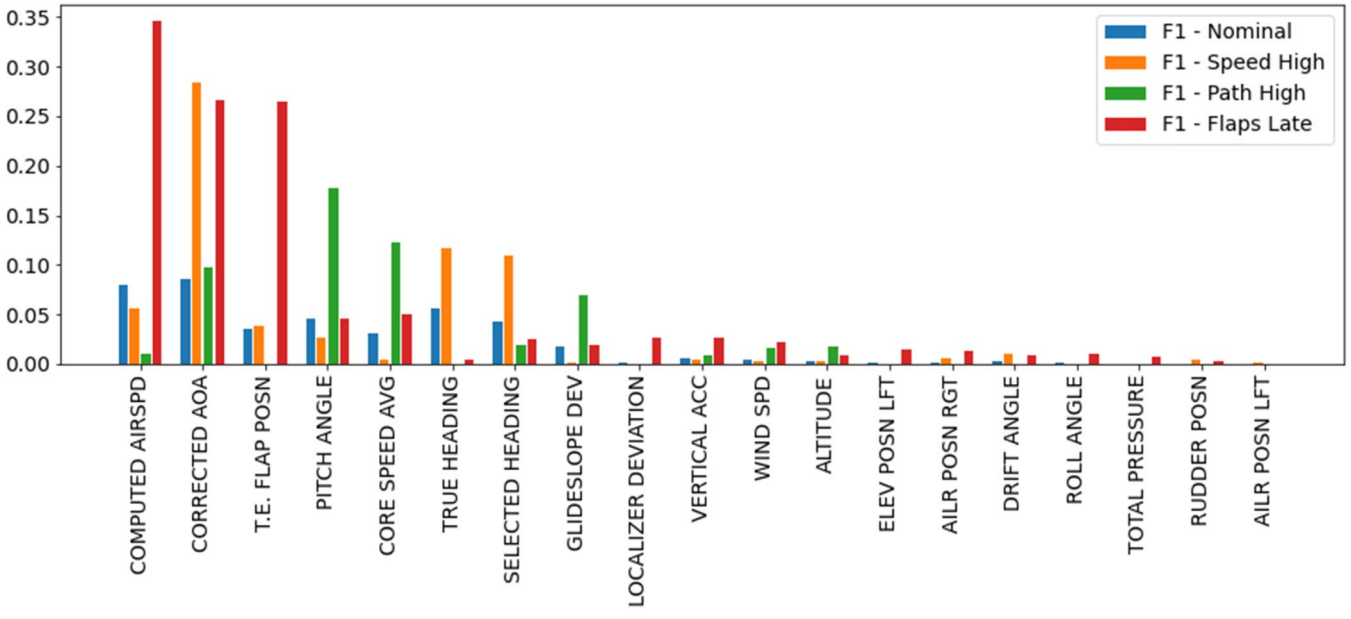


Fig. 7 Importance of parameters for multiple anomalies, as identified by the random permutation method.

problem as a case study. To be able to visualize the 256-dimensional latent space of the M1+M2 and CCLP models in two-dimensional (2D), we use t-distributed stochastic neighbor embedding (t-SNE) [25], implemented using the Python package Scikit-Learn, with the perplexity hyperparameter fixed at 50. Figure 8 compares the 2D visualizations of the 256-dimensional latent spaces of CCLP and M1+M2 color-coded by the class that each data point belongs to.

As mentioned before, the M1+M2 feature encoding is fully unsupervised, and as expected, the clusters that are formed in the latent space of this model do not necessarily correspond to the different classes that exist in the data. As a result, it is difficult to interpret such a latent space, and significant data postprocessing is needed to identify why and how different clusters have formed in the latent space. It should be noted that the reduction in dimensionality of the latent space from 256 to 2 also entails a loss of information on clustering (by class) present in the original feature space. On the other hand, the t-SNE-transformed latent space of the CCLP model exhibits compact clustering of the data according to class affiliation. This is not a surprising outcome because CCLP's objective function [Eq. (5)] enforces the compact clustering of data belonging to the same class via the label-propagation technique.

The visualization of CCLP's latent space reveals the interesting pattern of a central region where the four clusters (one per class) merge into one another. We were interested to see if such merging cluster actually form in the 256-dimensional latent space or if they are simply an artifact of t-SNE's nonlinear mapping from 256D to 2D. To examine this question, we applied agglomerative clustering (also implemented in Scikit-Learn) with five clusters (number of classes plus one) in the 256-dimensional latent space. The left panel of Fig. 9 shows the 2D visualization of the clusters formed in the latent space and shows that a fifth cluster (colored purple) forms exactly in the central region where the clusters of data belonging to different classes are merging. The intensity of the colors in the graph corresponds to the distance of each point to the centroid of the cluster that it belongs to, meaning that more transparent points are farther away from their clusters' centers.

After confirming that the data in the central region are being clustered together, we hypothesized that this central region is filled

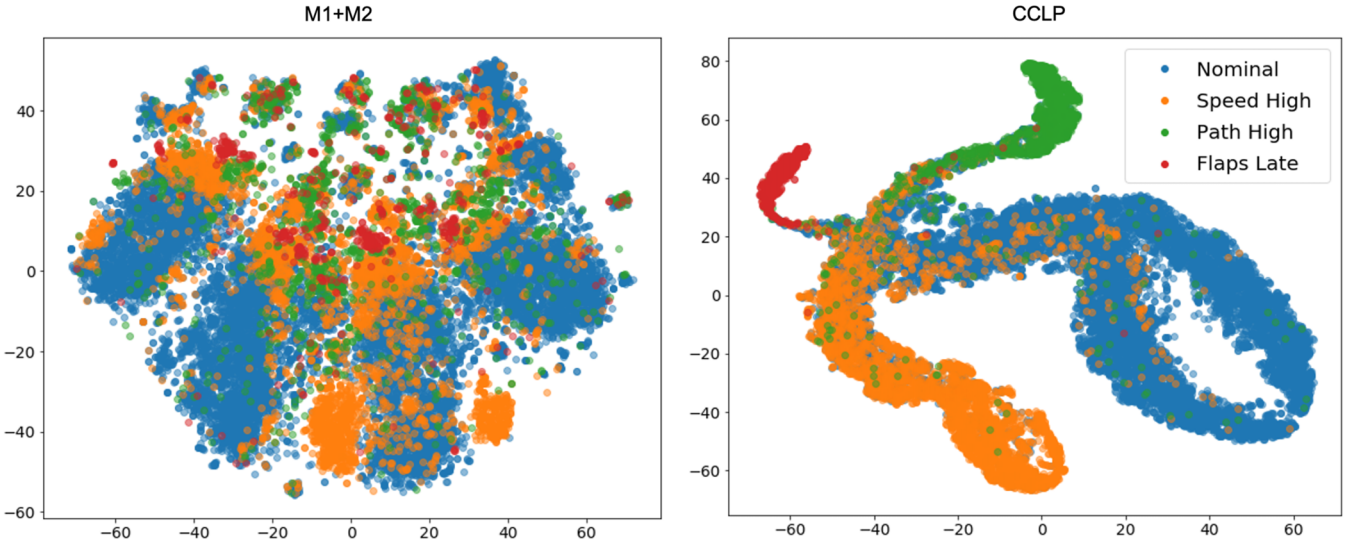


Fig. 8 Two-dimensional visualization of the 256-dimensional latent space of M1+M2 and CCLP using t-SNE, color-coded by the actual class of each data point.

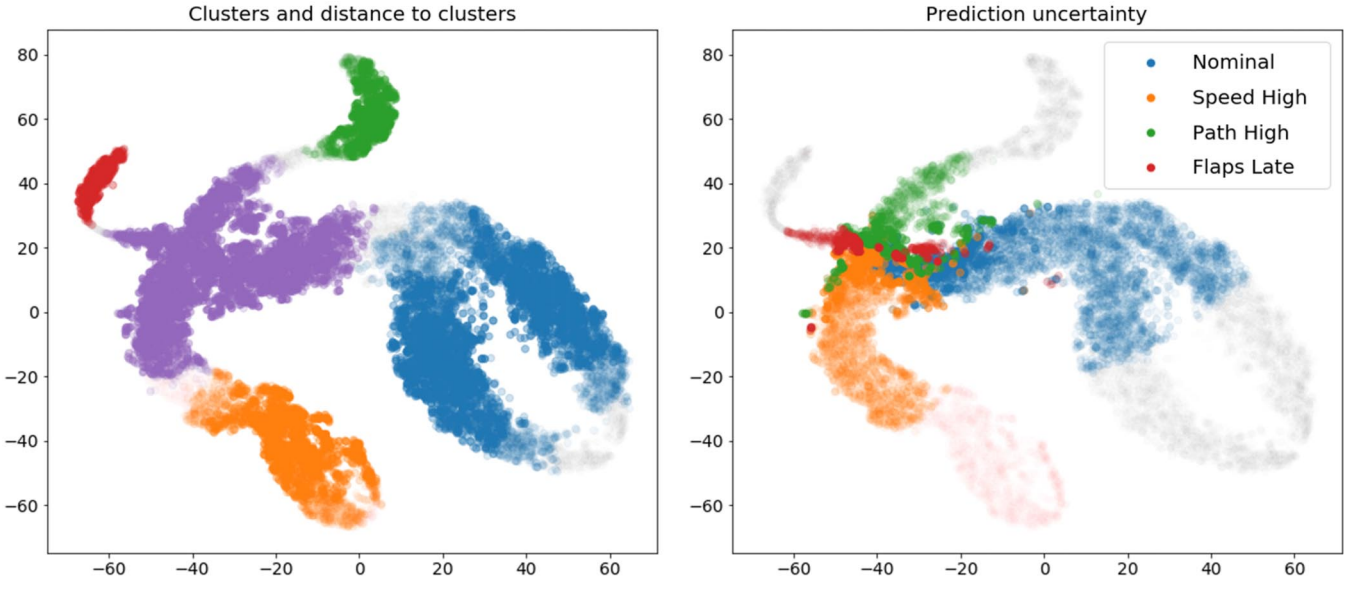


Fig. 9 Two-dimensional visualizations of the CCLP model's latent space, showing the five clusters formed in the latent space. The distance of data points to their clusters' centers (left), and the prediction uncertainty of the model's classifier (right).

with data points that are hard to classify. To validate this hypothesis, we visualized the prediction uncertainty of the CCLP classifier (right panel of Fig. 9). We quantified the prediction uncertainty of the model as the entropy of the output of the softmax layer of the classifier. The output of this layer is a four-dimensional vector, each component i of which denotes the probability that the data belong to class i, for $i \in \{1, 2, 3, 4\}$. The entropy is then given as

$$\mathcal{H}(X) = -\sum_{i=1}^{4} \hat{y}_i \log(\hat{y}_i) \tag{9}$$

where $\hat{y}$ is the classifier's prediction (output of softmax layer) for input X. In Fig. 9 the points that are higher in color intensity are associated with a higher prediction uncertainty (i.e., entropy). The Fig. 9 plots show that the majority of the points that are hard to classify belong to this central cluster (purple cluster in the left plot). On the other hand, data points that are easier to classify are compactly clustered away from this central cluster. These findings open up new avenues and ideas for designing active-learning strategies to identify and select the most valuable subset of unlabeled data points to be studied by subject matter experts. The feedback from subject matter experts' labels can help to improve prediction performance, by labeling the most informative hitherto unlabeled data.

V. Conclusions

The standard practice for anomaly detection in the aviation domain is exceedance detection, which is unable to identify complex anomalies as well as unknown risks and vulnerabilities. Supervised learning can partially overcome this challenge, but it is afflicted with requiring a high number of processed and labeled data points to reach optimum performance. However, in many real-world applications, such as aviation, labeled data are either not available or scarce. Moreover, supervised approaches do not address the challenge of detecting unknown vulnerabilities. As a result, the aviation anomaly-detection literature has mainly focused on unsupervised reasoning to identify anomalies in high-dimensional time series of flight data. Unfortunately, unsupervised learning, by nature, suffers from a high number of false alarms and low accuracy in complex settings. This limits its applicability. To address the challenges aviation data pose to both unsupervised and supervised reasoning, the authors developed a robust and more explainable semi-supervised model that takes advantage of the entirety of the available data (i.e., a majority of unlabeled data and a minority of labeled data) and identifies anomalies and precursors of potential safety incidents with high accuracy and a low number of false alarms. Our proposed model is a step toward development of a framework capable of identifying unknown risks and vulnerabilities in airspace operations. A future step is to extend the semi-supervised model to have open-set recognition capability, where the model can identify data as belonging to none of the known classes. Alternatively, active learning strategies can be deployed on data with high uncertainty to detect such unknown anomalies and have them reviewed and labeled by subject matter experts.

The authors deployed two successful deep semi-supervised classification models described in the machine learning literature, named M1+M2 [19] and CCLP [20], for anomaly detection in high-dimensional and heterogeneous time series of FOQA data. To validate these models and benchmark their performance, two case studies were developed: 1) binary anomaly detection during the takeoff phase of commercial aircraft and 2) multiclass anomaly detection during approach for landing of commercial aircraft. The performance of the two semi-supervised models was compared with the performance of the state-of-the-art supervised model (developed specifically for anomaly detection in aviation data) [7]. Although our case studies have focused on the takeoff and approach phases of flight, the developed model is equally applicable to any other flight phase.

Figures 3 and 5 show that both semi-supervised models significantly outperform the supervised model when trained on only a small number of labeled data points. For example, the CCLP model outperforms DT-MIL significantly (average accuracy of 72.2 and 31.4%, respectively) when only 100 data points (time series) are labeled, which is equivalent to 0.55% of the total data. Accuracy of CCLP reaches over 80% when 1000 labeled data points (5.5% of the total data) are available, whereas the supervised model (DT-MIL) performs below 60%.

In addition, the explainability of the semi-supervised models was investigated. Specifically, it is shown that the latent feature space of the CCLP model exhibits compact clustering of the data of different classes (Fig. 8, right panel). A very interesting discovery was made when correlating clusters formed in the latent space with uncertainty in classification (Fig. 9). It was found that the majority of the points that are hard to classify for the model belong to a central cluster (purple cluster in the left panel of Fig. 9) at the intersection of four clusters that have extensive overlap with the four classes investigated in this study. On the other hand, data points that are easier to classify are compactly clustered away from this central cluster. This observation opens up new avenues and ideas for identifying the most valuable subset of unlabeled data points to be labeled by subject matter experts, to improve future performance of the model.

Appendix: Additional Figures

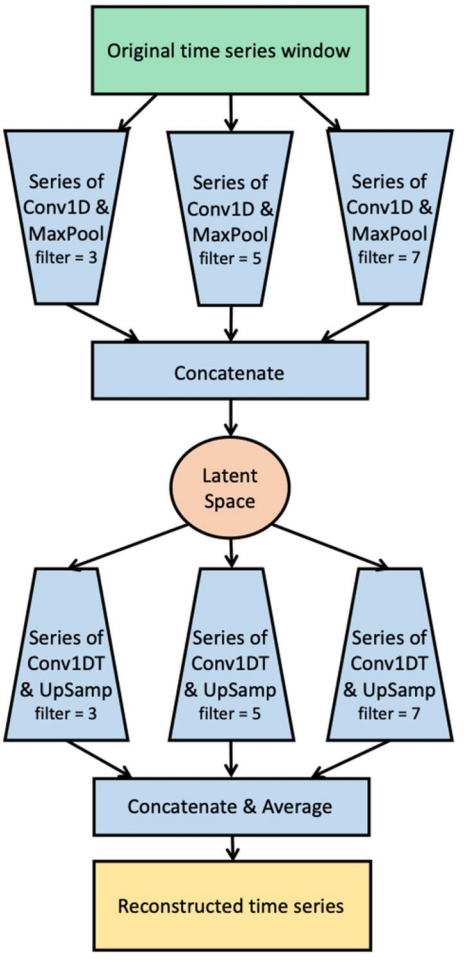


Fig. A1 Convolutional variational auto-encoder architecture [17] that is used for the M1 model in this paper.

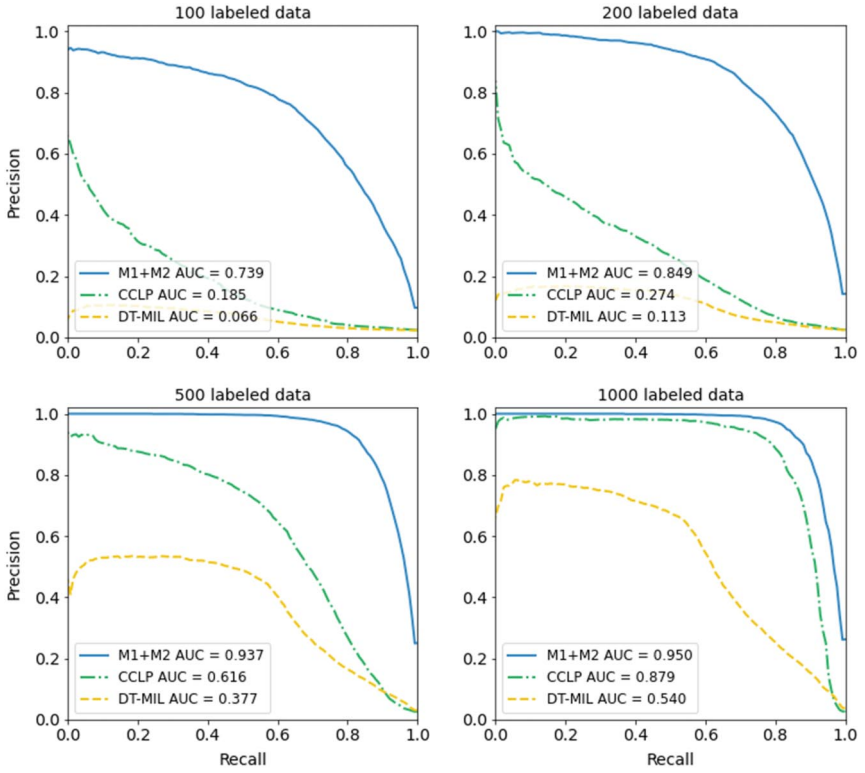


Fig. A2 Precision–recall curves and area under the curve (AUC) for performance of the M1+M2, CCLP, and DT-MIL models on the binary anomaly detection problem.

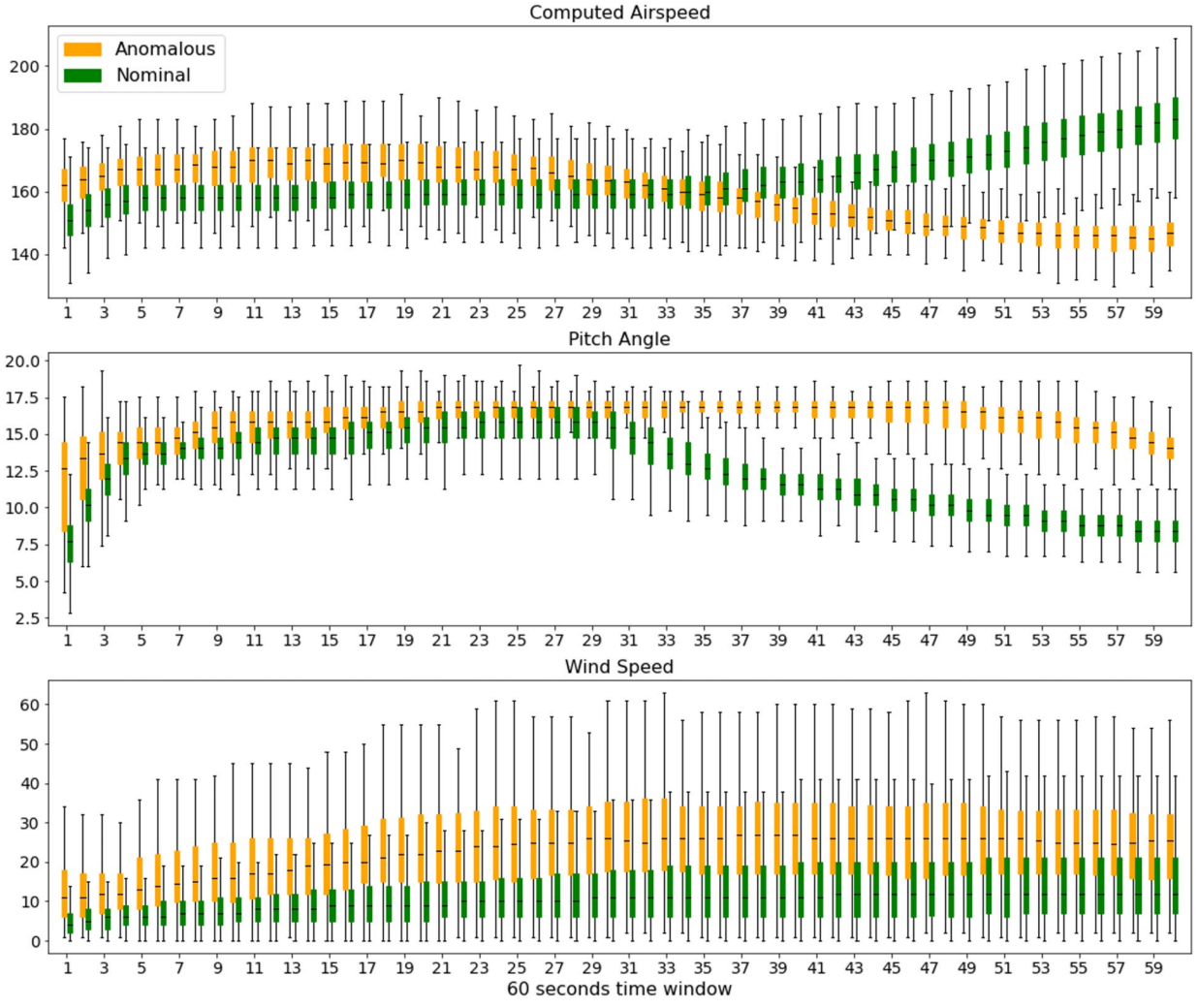
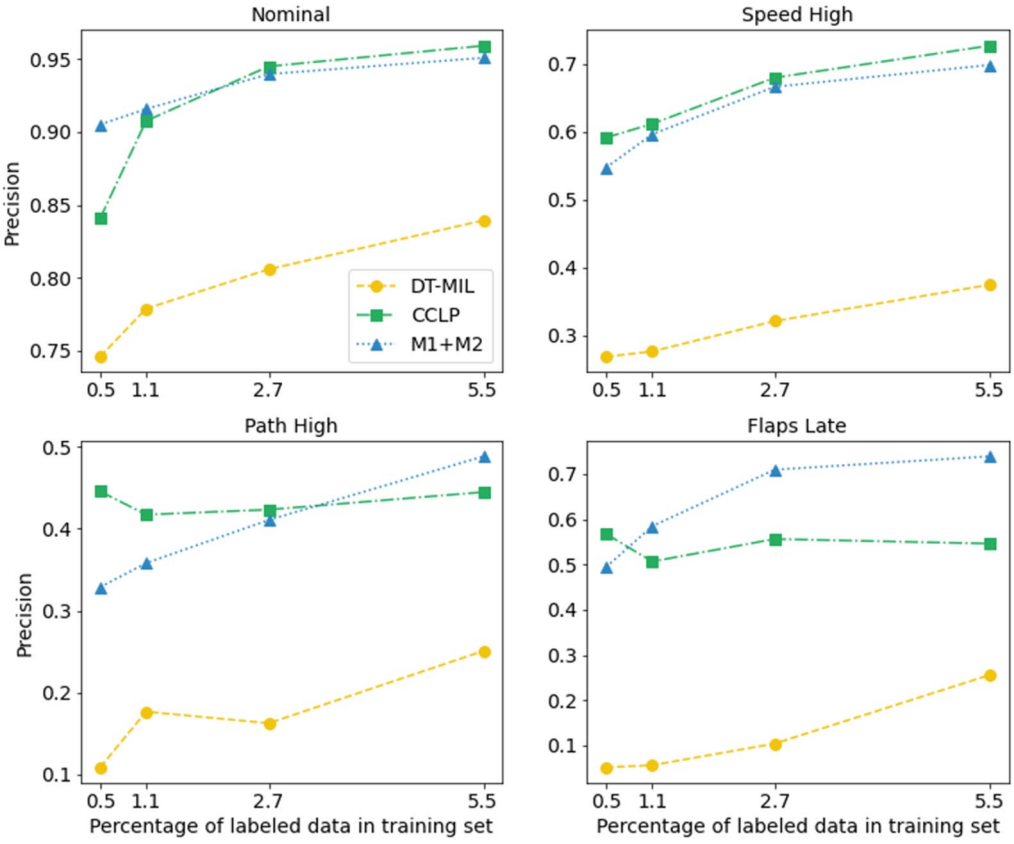
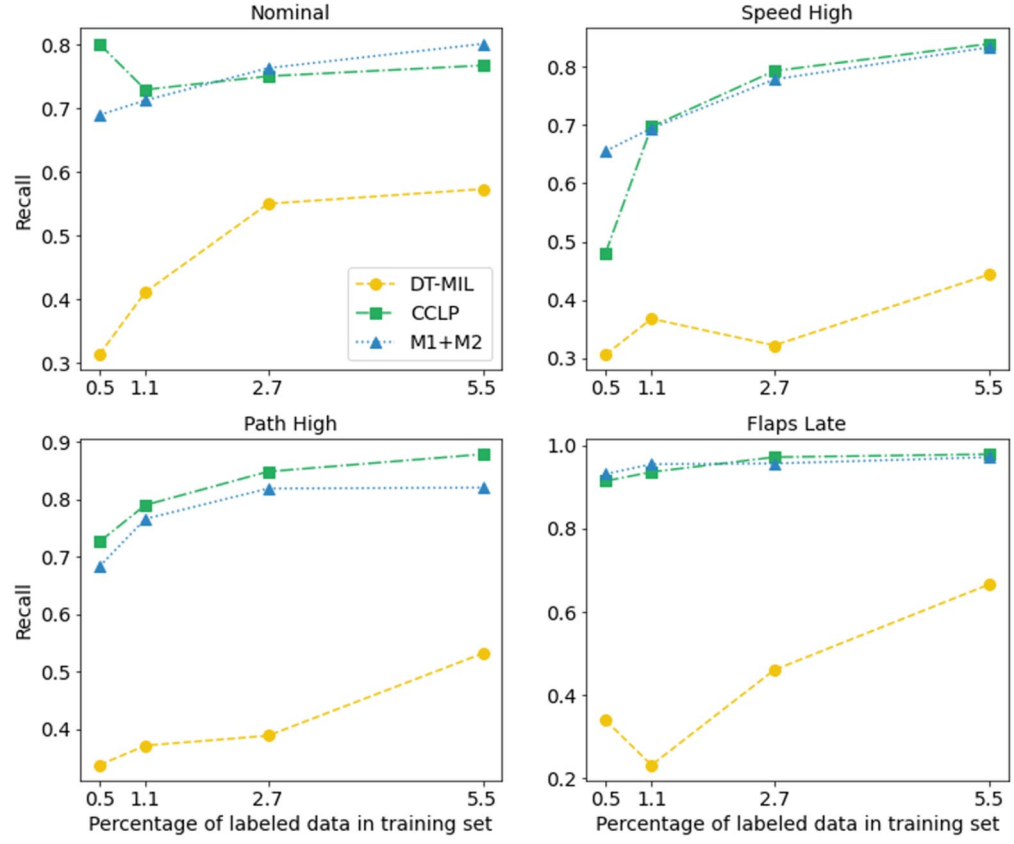


Fig. A3 Mean ± standard deviation of trajectories of top features for takeoff anomalies versus nominal.



a) Precisions per class



b) Recalls per class

Fig. A4 Precision and recall per class for semi-supervised (CCLP and M1+M2) and supervised (DT-MIL) models in the multiclass anomaly detection problem.

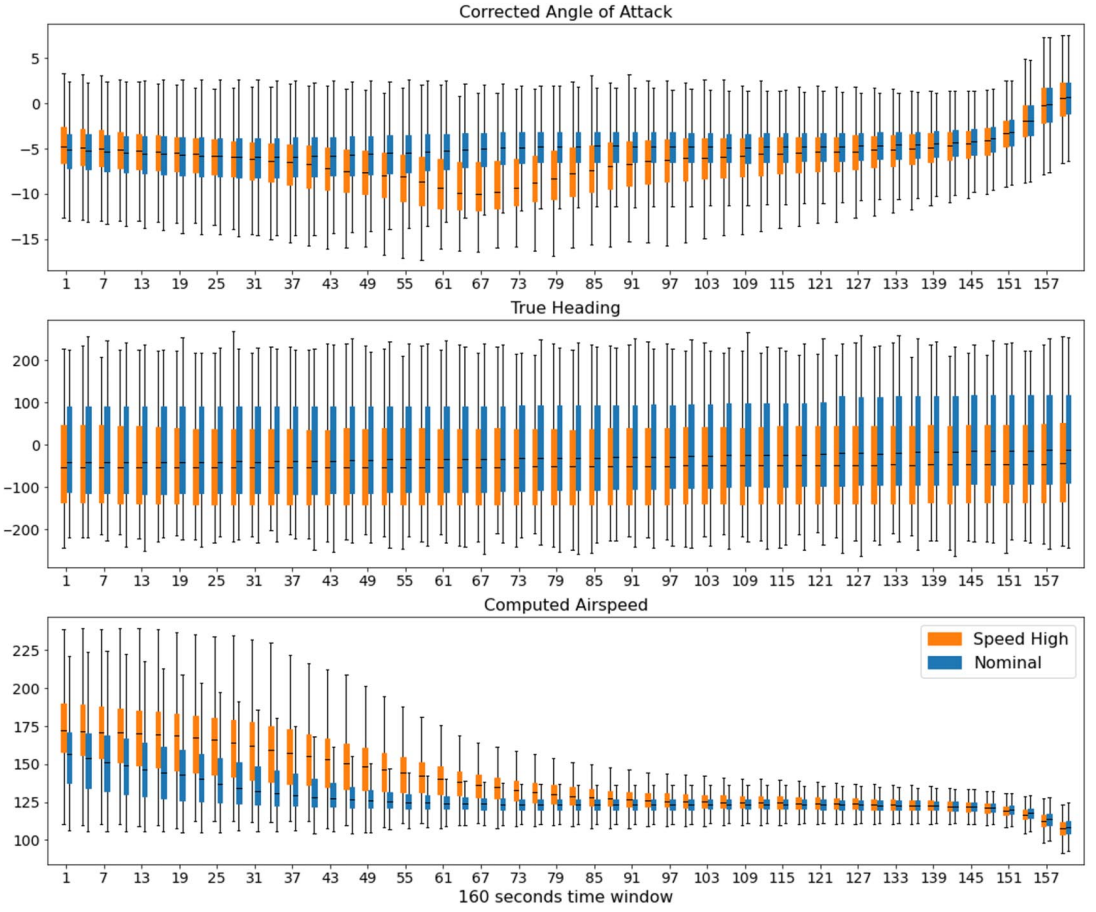


Fig. A5 Mean ± standard deviation of trajectories of top features for speed high anomalies versus nominal.

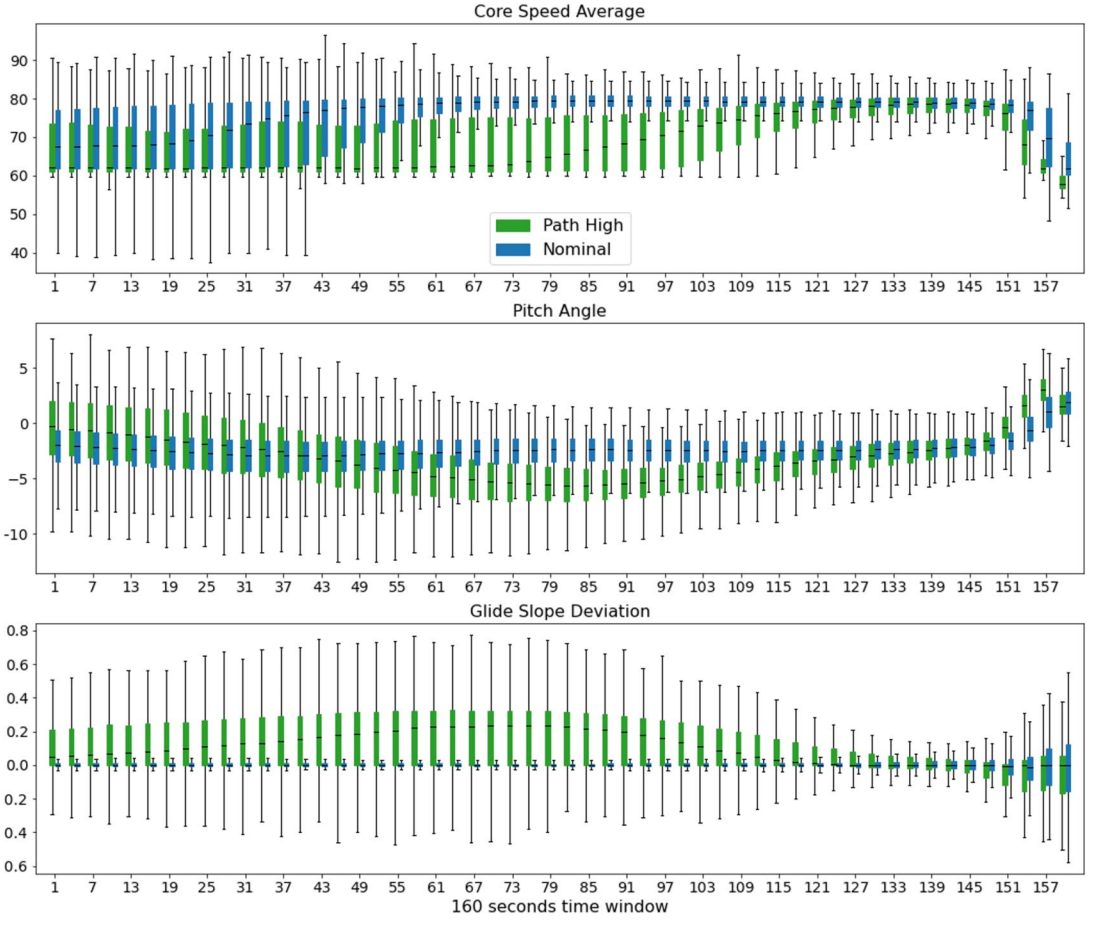


Fig. A6 Mean ± standard deviation of trajectories of top features for path high anomalies versus nominal.

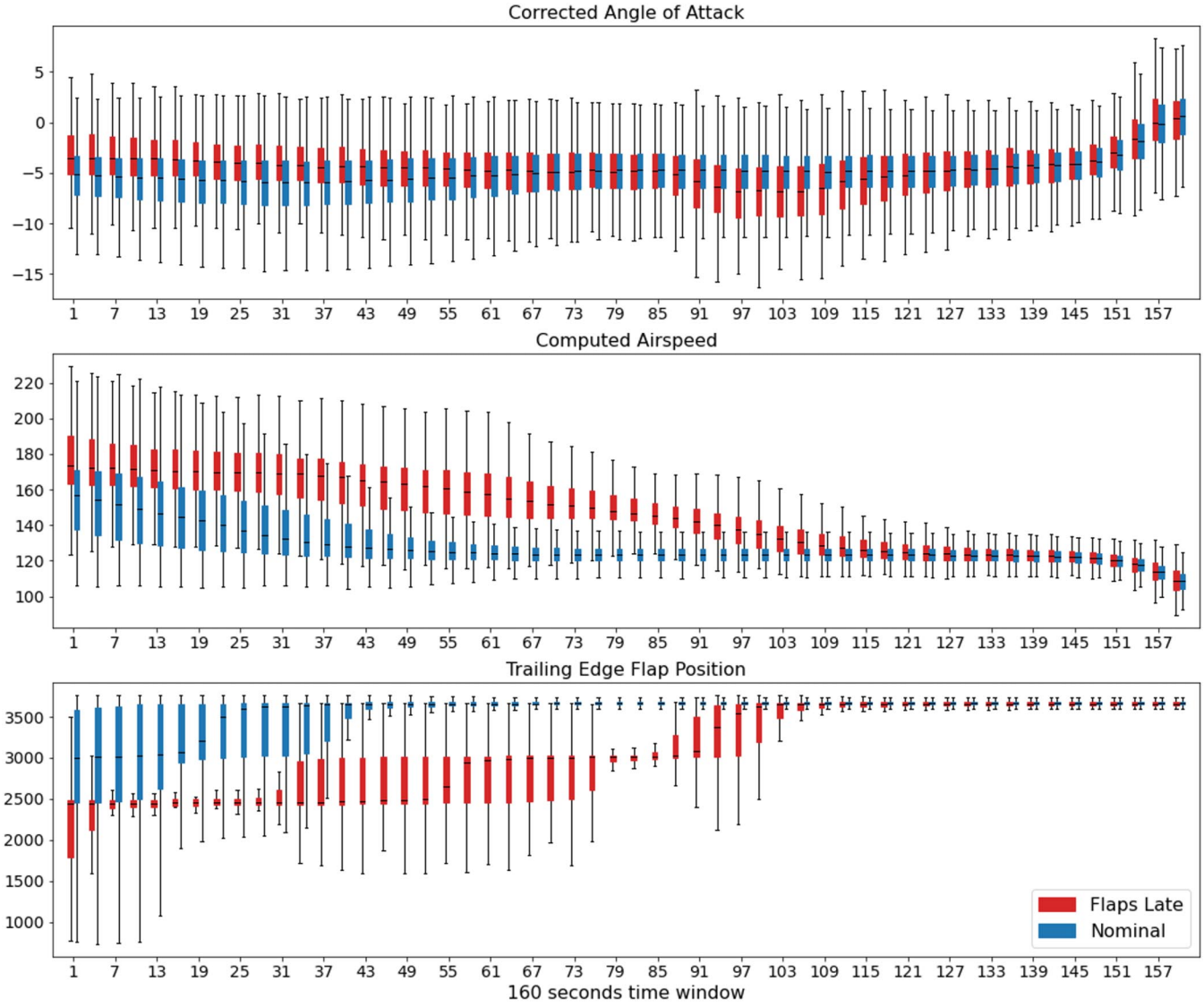


Fig. A7 Mean ± standard deviation of trajectories of top features for flaps late anomalies versus nominal.

Acknowledgment

The authors acknowledge the funding of this research from the NASA System-wide Safety Project under contracts 80ARC020D0010 and NNA16BD14C.

References

[1] "Annual Summaries of US Civil Aviation Accidents," National Transportation Safety Board, 2019, https://www.ntsb.gov/investigations/data/Documents/AviationAccidentStatistics_1999-2018_20191101.xlsx.

[2] "US Transportation Fatality Statistics," National Transportation Safety Board, 2017, https://www.ntsb.gov/investigations/data/Pages/AviationDataStats2017.aspx.

[3] Sprung, M. J., Chambers, M., and Smith-Pickel, S., "Transportation Statistics Annual Report 2018," United States Department of Transportation, Bureau of Transportation Statistics, 2018, https://rosap.ntl.bts.gov/view/dot/37861.
https://doi.org/10.21949/1502596

[4] "National Transportation Safety Board Aviation Investigation Manual Major Team Investigations," National Transportation Safety Board, 2002, https://www.ntsb.gov/investigations/process/Documents/MajorInvestigationsManual.pdf.

[5] "Flight Operational Quality Assurance," Federal Aviation Administration, TR No. 120-82, 2004, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_120-82.pdf.

[6] Lee, H., Madar, S., Sairam, S., Puranik, T. G., Payan, A. P., Kirby, M., Pinon, O. J., and Mavris, D. N., "Critical Parameter Identification for Safety Events in Commercial Aviation Using Machine Learning," *Aerospace*, Vol. 7, No. 6, June 2020, p. 73.
https://doi.org/10.3390/aerospace7060073

[7] Janakiraman, V. M., "Explaining Aviation Safety Incidents Using Deep Temporal Multiple Instance Learning," *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery (ACM), New York, 2018, pp. 406–415.
https://doi.org/10.1145/3219819.3219871

[8] Mori, R., "Anomaly Detection and Cause Analysis During Landing Approach Using Recurrent Neural Network," *Journal of Aerospace Information Systems*, March 2021 (to be published).
https://doi.org/10.2514/1.I010941

[9] Bay, S. D., and Schwabacher, M., "Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule," *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery (ACM), New York, 2003, pp. 29–38.
https://doi.org/10.1145/956750.956758

[10] Melnyk, I., Matthews, B., Valizadegan, H., Banerjee, A., and Oza, N., "Vector Autoregressive Model-Based Anomaly Detection in Aviation Systems," *Journal of Aerospace Information Systems*, Vol. 13, No. 4, April 2016, pp. 161–173.
https://doi.org/10.2514/1.I010394

[11] Iverson, D. L., "Inductive System Health Monitoring," *Proceedings of the International Conference on Artificial Intelligence*, CSREA Press, Las Vegas, NV, June 2004, https://ti.arc.nasa.gov/m/groups/intelligent-data-understanding/ICAI2004-Iverson.pdf.

[12] Budalakoti, S., Srivastava, A. N., and Otey, M. E., "Anomaly Detection and Diagnosis Algorithms for Discrete Symbol Sequences with Applications to Airline Safety," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 39, No. 1, 2009, pp. 101–113. https://doi.org/10.1109/TSMCC.2008.2007248

[13] Das, S., Matthews, B., Srivastava, A. N., and Oza, N., "Multiple Kernel Learning for Heterogeneous Anomaly Detection: Algorithm and Aviation Safety Case Study," *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery (ACM), New York, 2010, pp. 47–56. https://doi.org/10.1145/1835804.1835813

[14] Matthews, B., Srivastava, A. N., Schade, J., Schleicher, D., Chan, K., Gutterud, R., and Kiniry, M., "Discovery of Abnormal Flight Patterns in Flight Track Data," *Proceedings of 2013 Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2013-4386, 2010. https://doi.org/10.2514/6.2013-4386

[15] Lee, H., Li, G., Rai, A., and Chattopadhyay, A., "Real-time Anomaly Detection Framework Using a Support Vector Regression for the Safety Monitoring of Commercial Aircraft," *Advanced Engineering Informatics*, Vol. 44, April 2020, Paper 101071. https://doi.org/10.1016/j.aei.2020.101071

[16] Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T., "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery (ACM), New York, 2018, pp. 387–395. https://doi.org/10.1145/3219819.3219845

[17] Memarzadeh, M., Matthews, B., and Avrekh, I., "Unsupervised Anomaly Detection in Flight Data Using Convolutional Variational Auto-Encoder," *Aerospace*, Vol. 7, No. 8, Aug. 2020, p. 115. https://doi.org/10.3390/aerospace7080115

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

[19] Kingma, D., Rezende, D., Mohamed, S., and Welling, M., "Semi-Supervised Learning with Deep Generative Models," *Advances in Neural Information Processing Systems (NeurIPS)*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., 2014, https://arxiv.org/abs/1406.5298.

[20] Kamnitsas, K., Castro, D., Le-Folgoc, L., Walker, I., Tanno, R., Rueckert, D., Glocker, B., Criminisi, A., and Nori, A., "Semi-Supervised Learning via Compact Latent Space Clustering," *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 2459–2468, https://arxiv.org/abs/1806.02679.

[21] Chapelle, O., Scholkopf, B., and Zien, A., *Semi-Supervised Learning*, MIT Press, Cambridge, MA, 2006, pp. 5–6, Sec. 1.2.2.

[22] Kingma, D., and Welling, M., "Auto-Encoding Variational Bayes," *International Conference on Learning Representations (ICLR)*, 2013, https://arxiv.org/abs/1312.6114.

[23] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A., "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *International Conference on Learning Representations (ICLR)*, 2017, https://openreview.net/forum?id=Sy2fzU9gl.

[24] Kingma, D., and Ba, J., "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, 2014, https://arxiv.org/abs/1412.6980.

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

C. Torens
Associate Editor