Multi-Class Anomaly Detection in Flight Data using Semi-Supervised Explainable Deep Learning Model

Milad Memarzadeh*, Bryan Matthews†, and Thomas Templin‡
Data Sciences Group, NASA Ames Research Center, Moffett Field, CA 94035, USA

Identifying precursor for 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's behavior and alarm exceedance from such safety margins. However, this approach is incapable of identifying unknown risk and vulnerabilities. Machine learning has been long studied and deployed to identify precursors for such anomalies, with the great challenge of the need for sufficient labelled set of data to achieve a reliable and accurate performance. In this article, we develop an explainable deep semi-supervised model for anomaly detection in aviation, building upon recent advancements in the machine learning literature. The proposed model combines feature engineering and classification in the feature space, while leveraging all available data (labelled and unlabeled). Validating on two case studies of anomaly detection in take-off and landing phases of commercial aircraft, we show that our model is able to outperform state-of-the-art supervised anomaly detection model and reach significantly high accuracy and low false alarm with minimum amount of available labelled data.

I. Nomenclature

X, X_L, X_U	=	input data, labelled data, and unlabeled data
N_L, N_U	=	total number of labelled data, and unlabeled data.
y_L	=	labels for the labelled set
$\hat{y}$	=	classifier's prediction/output
z, w	=	variables representing the latent feature space
ϕ, ϕ'	=	parameters of encoder models
θ, θ'	=	parameters of decoder models
ψ	=	parameters of the classifier models
β	=	hyper-parameter controlling the effect of KL-divergence in the M1 model
α	=	hyper-parameter controlling the effect of classification loss in the M2 model
η	=	hyper-parameter controlling the effect of compact clustering loss in the CCLP model
T, H	=	optimal and estimated transition functions in the CCLP's label propagation
TP, TN, FP, FN	=	true positive, true negative, false positive, and false negative
$\mathcal{H}$	=	entropy

II. 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 trending success can be attributed to both improving automation with redundant hardware and software protections, as well as an increased focus on proactive monitoring and response to real-time and historically identified vulnerabilities, by implementing more resilient procedures and protocols. While the number of passenger enplanements has increased 20% from 706 million in 2009 to 851 million in 2017, the number of departures has decreased 5% from 9.7 million to 9.3 million over the same period [2]. This trend has resulted in a historically high passenger load factor of 82.3% in

*Senior Scientist, USRA, Corresponding Author: milad.memarzadeh@nasa.gov
†Research Engineer, KBRwyle.
‡Research Computer Scientist, NASA Ames Research Center.

2017 [3]. As this factor steadily increases, it is expected the number of departures will begin to rise in the near future. To remain at this historically low level of accidents per year, the NAS will need to develop and proactively identify previously unknown operationally significant safety events occurring in the daily operations.

Identifying situations where unknown risk or vulnerabilities exist is not a trivial problem. Much of the knowledge of adverse events comes from after-the-fact analysis using a forensic approach to investigations to determine the root cause of an incident or accident such as the process that NTSB uses when investigating accidents [4]. This area is where machine learning can improve the process and eventually help with risk mitigation in the form of identifying safety requirements that need revising to address the an emerging vulnerability. However, the events that are automatically identified still need to be reviewed and assessed by subject matter experts familiar with the procedures. 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 contributing factors with proper countermeasures. These may include improved pilot/controller training or developing automation safety procedures. If these measures are put in place they can help to avoid states that result in an increased likelihood of an incident or accident that may result in damage to the aircraft, injury, or loss of life.

One of the main challenges of identifying precursors to safety events using machine learning in aviation domain is the sparsity and quantity of processed and labelled data, where such precursors are reliably identified and labelled. As a result, majority of the development in the literature focus on unsupervised reasoning of the data to identify anomalies in high-dimensional time series of flight. Unfortunately, due to the nature of unsupervised learning, majority of such methods suffer from high number of false alarms and low accuracy in complex settings, which limits their applicability. On the other hand, supervised reasoning of the data cannot reach the optimum performance due to the lack of labelled data. In order to address the challenges of both unsupervised and supervised reasoning of aviation data, in this article we develop a robust and explainable semi-supervised model that takes advantage of the entirety of the available data, the majority unlabeled data and the minority labelled data, and identify precursors for anomalies with high accuracy and low number of false alarms.

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

III. Related Work

The main approach for identifying vulnerabilities in operations leverages domain expertise about how the system should behave within the expected tolerances to known safety margins. This approach is known as exceedance detection, which is the standard technique for anomaly detection in aerospace data [5]. This technique compares specific parameters against pre-defined 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.

In order to identify unknown risks and vulnerabilities, we need to go beyond simplistic approaches. Recent advancements in the field of machine learning suggest the application of machine learning techniques for identifying anomalies in aviation data. Generally speaking, machine learning approaches used for aviation anomaly detection can be categorized into supervised and unsupervised methods, with the presence or absence of labelled data the key differentiator between the two. If the data is reviewed by the subject matter experts after the fact, usually the location and the type of anomaly that is present in the data can be identified; this is the definition of the labelled data. On the other hand, due to the huge volume of recorded data in the aviation, majority of the available data is not reviewed and hence labelled; this is what we define as unlabeled data.

Supervised learning models only build their inference using the labelled data and have demonstrated impressive performance when trained on a sufficient number of labeled data. Lee et al. [6] developed uses several classic supervised learning approaches to identify safety anomalies in the FOQA data. Janakiraman [7] developed a deep temporal multiple instance learning (DT-MIL) approach by taking advantage of recurrent neural networks for supervised classification of adverse events in FOQA data. Although both of these models exhibit great performance, for many real-world problems, including aviation domain, labeling data requires a great effort from subject matter experts and is largely impractical. As a result, the size of reliable labelled set of data is not large enough for such supervised models to reach optimum performance.

Due to difficulty of obtaining labels for aviation data, another area of development has been around applications of unsupervised learning approaches for anomaly detection in aviation. Descent variety of methods have been developing in this area including proximity based methods [8], clustering based methods [9, 10], kernel based methods [11], and deep learning based methods [12]. All of these methods use different techniques to identify the majority nominal pattern in the unlabeled pool of data, and then use a distance metric to identify data that are far away from the majority pattern. Although unsupervised models address the problem of lack of labelled data, they suffer with high number of false alarms and low accuracy of anomaly detection, especially in complex data such as high-dimensional heterogeneous time series of flight data. This is typically due to the fact that statistically significant anomalies are not guaranteed to always be operationally significant anomalies. For example, we showed in our previous study that performance of the unsupervised model increased by $\sim 32\%$, when the model was only trained on the majority nominal data [12], meaning that incorporation of a weak supervision (removing anomalies from training) had a significant impact on the model's performance.

Motivated by the above finding, we propose to develop semi-supervised learning [13] that can leverage availability of both majority unlabeled data and minority labelled ones. semi-supervised models usually combine unsupervised feature engineering (based on all available data) with supervised classification (based on labeled data). Figure 1 shows illustrates the overall semi-supervised model proposed here. The main hypothesis is that by leveraging the large pool of unlabeled data, unsupervised feature engineering can be improved so that supervised classification in feature space can reach optimal performance. We specifically build upon our previous unsupervised feature engineering model, Convolutional Variational Auto-Encoder (CVAE) [12], and two successful deep semi-supervised classification models developed in the machine learning literature, named M1+M2 [14] and Compact Clustering via Label Propagation (CCLP) [15], for anomaly detection in high-dimensional and heterogeneous time series of FOQA data.

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

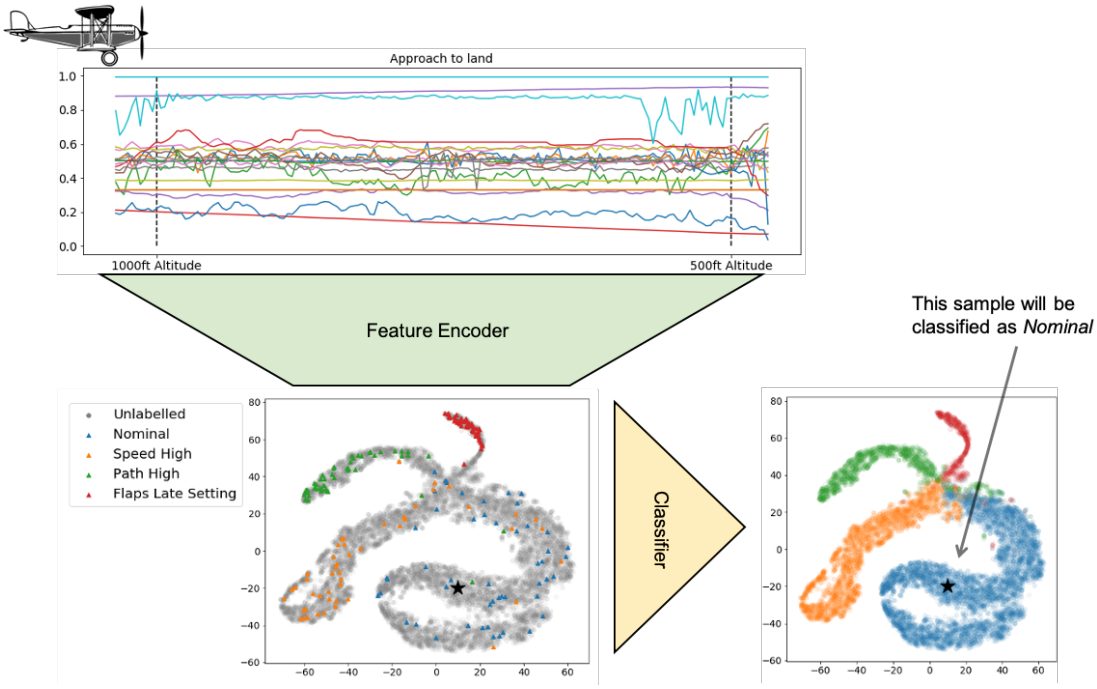

Fig. 1 Graphical illustration of the proposed semi-supervised anomaly detection approach. The latent space configurations in this figure are from CCLP approach.

IV. Method

In this section, we summarize two semi-supervised models that are adopted for the purpose of anomaly detection. Let us assume that the available data is grouped in two sets of labelled 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 before, any unsupervised learning ignores y_L, while supervised learning ignores X_U. The goal of semi-supervised model is to not only utilize both sets of data, but to make sure that deployment of both X_U and y_L improves the performance compared to unsupervised and supervised models [13]. In the next sections, we summarize two models: (1) M1+M2 model [14], which is a deep generative model built upon recent success in deploying Variational Auto-Encoder (VAE) [16], and (2) compact clustering via label propagation (CCLP) model [15], which is a deep encoding model that enforces compact clustering of the data belonging to the same class in the latent feature space. As illustrated in Figure 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 is anomalous or not, and if there are multiple anomalies, classifies the anomaly type. We take advantage of our previously developed CVAE model architecture [12] for feature encoding in both of these semi-supervised models. The main difference between the two models is in the objective function 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, Deep Temporal Multiple Instance Learning (DT-MIL) [7], which utilizes deep recurrent neural network to take temporal dependency of the data into account and is specifically developed for anomaly detection in aviation data. We benchmark the performance of these models using two sets of Flight Operational Quality Assurance (FOQA) data, (1) a data that consists of one type of anomaly during the take-off phase of commercial aircraft, and (2) a data that consists of multiple types of anomalies during the approach to landing of commercial aircraft. We will discuss details of these data later on.

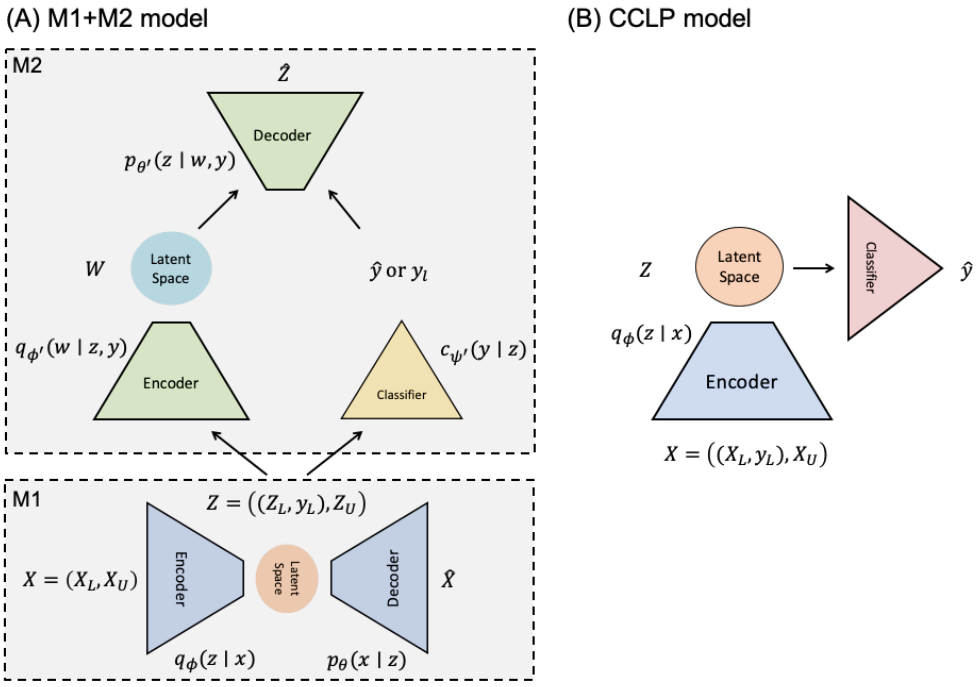

Fig. 2 Graphical illustration of (A) M1+M2 and (B) CCLP model architectures.

A. M1+M2: semi-supervised deep generative model

As depicted by name, M1+M2 model [14] is a combination of (i) a M1 model, which is a β-VAE [17] and serves mainly as the unsupervised feature encoder and maps the input data to a lower-dimensional latent space, and (ii) a M2 model, which is a VAE deployed in the latent space of M1 model with addition of a classifier. Figure 2A illustrates M1+M2 model graphically. These two models are trained separately.

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:

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

where the first term is the log likelihood of the reconstructed data conditioned on the samples from the latent space, and the second term is the KL-divergence between the prior distribution of the latent space ($p_\theta(z)$), which is assumed as standard Normal distribution, and the posterior distribution of the latent space ($q_\phi(z \mid x)$). β is a hyper-parameter controlling the importance of the KL-divergence term in the above objective function. $q_\phi(z \mid x)$ and $p_\theta(x \mid z)$ could be any function that serve as encoder and decoder respectively, however, in this article we use a deep neural network to represent these functions and ϕ and θ are the parameter (or weights) of the neural networks.

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

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

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

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

The second term is the classification loss, which is defined as cross entropy and ψ is the parameters (or weights) of the classifier, and α is a hyper-parameter that denotes reliance of the labelled data. Moreover, the third term is the VAE objective function for the unlabeled set and is defined as follows:

$$\mathcal{U}(z) = \sum_{y} c_\psi\left(y \mid z\right) \mathcal{L}(z, y) - \mathcal{H}\left(c_\psi\left(y \mid z\right)\right) \tag{4}$$

where the first term marginalizes over all possible labels, and the second term forces the classifier to make confident predictions for the unlabeled data, so that it can improve the reconstruction.

B. CCLP: semi-supervised model via compact latent space clustering

Compact Clustering via Label Propagation (CCLP) approach [15] follows the cluster assumption in semi-supervised machine learning [13], which 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, which 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 embedding of labelled and unlabeled samples to capture underlying structure in latent feature space and to use label propagation to estimate its high- and low-density regions. This idea is enforced by 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:

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

Where the first term is the classification loss (cross entropy) for the labelled set, the second term is the cross entropy between the optimal transition function T and the estimated one via label propagation H, and η is the hyper-parameter tuning the importance of this term in the whole objective function. The optimal transition function, T, denotes a desired optimal state in the latent feature space, where a single, compact cluster is formed per class in the data. In such 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 can be found as follows:

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

Where $N = N_L + N_U$ is the total number of samples, H is the transition matrix of the estimated structure of data via dynamic graph construction and label propagation, and $H^{(s)}$ is the transition function for s steps. For further details regarding the computation of the optimal and estimated transition functions, T and H, refer to [15].

V. Results and Discussion

In this section, we first note the details of the models that are implemented here. Next, we introduce two sets of data for anomaly detection during take-off and approach to landing of commercial aircraft and evaluate the performance of the semi-supervised models (M1+M2 and CCLP) compared 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 post-processing.

A. Implementation Details

Both of semi-supervised models introduced here are implemented in Python using PyTorch library*, while the supervised DT-MIL model is used as it was implemented using Keras library†. Although they were implemented using different libraries, this difference will not affect their performance. All of the models are trained with sufficient number of epochs and have been reviewed to confirm the convergence of the model parameters. The `Adam` optimizer [18] is used for all of the models listed in the following subsections.

1. M1+M2 model

As noted before, M1+M2 model consists of two separate models: (1) M1 model, which serves as unsupervised feature engineering role, mapping the input data to the latent feature space, and (2) M2 model, which serves primarily as the classification of the data (nominal or anomalous, or what type of anomaly in the case of multi-class anomaly detection). It should be noted that M2 model is superior to simply adding a classifier in the latent space of the M1 model as shown by the original paper [14], since it adds extra terms in the objective function forcing the classifier to make confident predictions, which helps the reconstruction of the latent feature space with lower error.

We have used our previously developed CVAE model architecture [12] to serve as the M1 model in this article. The details of the model's architecture is illustrated in Figure 10 in the Appendix. The hyper-parameter β in Eq. (1) is set to 0.001, and we use binary cross entropy to estimate the likelihood of the reconstructions (i.e., first term in Eq. (1)). The latent space dimension of the M1 model is fixed to 128-dimensions for the binary problem and to 256-dimensions for the multi-class problem. The reason for the increase in the dimensionality of the latent space from binary problem to the multi-class one is the that the dimension of the multi-class problem's input data is much higher than the binary one (we will discuss these data in the next section).

M2 model is implemented using a fully connected architecture. The encoder consists two fully connected layers with 100 neurons each and `relu` activation that connects to the `linear` layer with the number of neurons equal to the dimension of the latent space. The decoder is also identical to the encoder, but in reverse order. The classifier has two fully-connected layers with 100 neurons each and `relu` activation with a `dropout` layer (50%) between them, which then connects to the output layer with 1 neuron with `sigmoid` activation for the binary problem and neurons equal to number of classes for multi-class problem with `softmax` activation. The hyper-parameter α in Eq. (2) is fixed to $0.1 \times N_U$ for all experiments, where N_U is the size of unlabeled set.

2. CCLP model

CCLP model consists of two parts: (1) an encoder, which is implemented identical to the encoder of M1 model (illustrated in Figure 10 in the Appendix), and (2) a classifier, which is identical to the classifier of M2 model in the previous subsection. Hyper-parameter η in Eq. (5) is fixed to 1 and the number of steps in the transition function, S in Eq. (6) is fixed to 3, both according to the original developers' recommendation and experimentation [15].

3. DT-MIL model

Architecture of the DT-MIL model is identical to the one that was constructed by the original developer [7]. It consists of a `GRU` layer with five neurons and `tanh` activation, followed by `time-distributed` fully connected layer

*`https://pytorch.org/`

†`https://keras.io/`

of 500 neurons with `tanh` activation, connected to a `time-distributed` fully connected layer with 1 neuron and `sigmoid` activation, which calculates the precursor scores for each time-stamp in the data. Then the result of the precursor layer is pushed through a global `max-pooling` layer to classify the data.

As it can be noted, DT-MIL is a binary classification model. Since we did not intend to alter the original model, for the case of multi-class problem, we implement DT-MIL in one-vs-all scheme for multi-class classification.

B. Binary Anomaly Detection in Take-off

In this problem, we focus on identifying one type of anomaly during take-off of the commercial aircraft. The precursor for this anomaly is identified by extensive reviews of the data by subject matter experts and is defined as a threshold on the drop of airspeed during the first 60 seconds of the take-off. If the airspeed drops more than 20 knots in the first 60 seconds of the take-off, it could lead to an undesirable incident. Based on this insight, we have designed a binary anomaly detection data set to test different models on how accurate they can identify this specific type of anomaly during the take-off phase. This is not necessarily the only type of anomaly in the data set and the true number of operationally significant safety anomalies are unknown; however, our objective is to measure the effectiveness of the algorithms and are using this particular safety incident as one measure of the algorithm's performance.

As a result, we have pre-processed FOQA data to set up training/testing data sets for bench-marking semi-supervised models (M1+M2 and CCLP) compared to the state-of-the-art supervised model (DT-MIL). Each data sample is a 60-seconds recording of multi-variate time series of 19 variables measuring the roll attitude (roll angle), altitude, pitch attitude (corrected angle of attack (AOA) and pitch angle), speed information (computer airspeed, low rotor speed (N1 actual) and high rotor speed (N2 actual), yaw attitude (drift angle, rudder position, and wind speed), control surface variables (ailerons positions, elevator positions, and stabilizer position) and flaps configurations (deployed at 10 degrees (Flaps0), 15 degrees (Flaps1), 20 degrees (Flaps2) and 40 degrees (FlapsFull)). It should be noted that FOQA data contain hundreds of variables, and these 19 variables used in our study were selected due to their relevance to the take-off phase of flight based on the guidance of subject matter experts.

1. Training/Testing Data Setup

The training data consists of 16407 samples, among which only 402 are anomalous (roughly 2.5%), and similarly the testing data consists of 5470 samples, among which 126 are anomalous. The training data is used for training each of the models and the testing is used to estimate the unbiased performance of the model. All the results in the figures are based on the performance of the models on the testing data. In order to evaluate the performance of these models on anomaly detection we assume that the nominal data is the *negative* class and the anomalous data is the *positive* class. As a result, two important metrics can be calculated as follows:

$$\begin{aligned} \text{precision} &= \frac{TP}{TP+FP} \\ \text{recall} &= \frac{TP}{TP+FN} \end{aligned} \tag{7}$$

where TP is true positive: anomalies that are correctly identified, FP is false positive (or false alarms): anomalies that are incorrectly identified, and FN is false negative, or anomalies that are missed and classified as nominal by mistake. Higher precision means that the model generates lower number of false positives, while higher recall means that the model miss-classifies fewer number of anomalies.

Although, we have labels available for the entire training data set, for the sake of experimentation, we evaluate the performance of the models when only small portion of the entire training set is labelled. As a result, we divide the training set into to sets of labelled data, (X_L, y_L), and unlabeled data, X_U, where $N_L + N_U = N = 16407$. Figure 3 compares performance of the three models in terms of precision (top panel) and recall (bottom panel), when the size of labelled set, N_L, is equal to 100 (0.6% of training data), 200, 500, and 1000 (6% of training data), respectively. The labelled set is randomly sampled from the pool of the training data and due to the nature of the randomness, we repeat the training for each model 50 times and the bars show mean $+/-$ standard deviation of the performance on the testing data based on these 50 independent experiments.

As illustrated by the figure, both semi-supervised models outperform the supervised model both in precision and recall, while M1+M2 performs consistently better than CCLP in this problem. It is quite interesting to see that on average, M1+M2 is able to reach nearly 97.5% precision and 70% recall with only 500 labelled data (3% of the total

training set). It should be noted that among these 500 labelled data, there are only 2.5% or $\sim$ 12 anomalous examples, since the pool of labelled data is selected randomly. This shows an incredible performance of the M1+M2 model in this case study, that is able to identify anomalies with high precision and recall with minimally labelled data. We can also infer that, as the number of labelled data increases, the performance of CCLP model gets closer to M1+M2 model, while the supervised model is significantly less accurate compared to the two semi-supervised models and it performs very poorly in the case of 100 and 200 labelled sets.

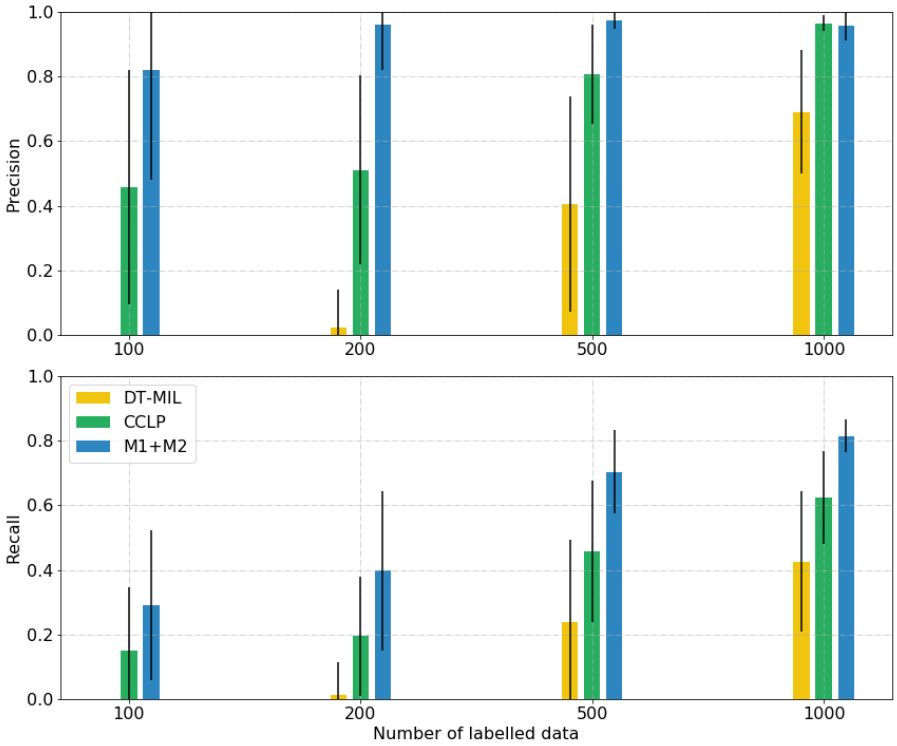

Fig. 3 This figure shows the performance of the semi-supervised models (M1+M2, and CCLP) and supervised model (DT-MIL) on the binary anomaly detection problem.

Figure 4 shows the importance of the features in both precision and recall in the anomaly detection performance. We have used random permutation method to obtain this figure. As illustrated, the most important features that help identifying the anomalies are the computed airspeed, pitch angle, and the wind speed. This result is not unexpected since the precursor for anomalies are identified based on the drop in the airspeed and these factors are all recording information relevant to the speed of the aircraft. As a point of comparison and to better understand how these features are helping the classification of anomalies, Figure 12 in the Appendix illustrates the mean $+/-$ standard deviation of the trajectory of these top three variables for nominal versus anomalous data examples in the testing data.

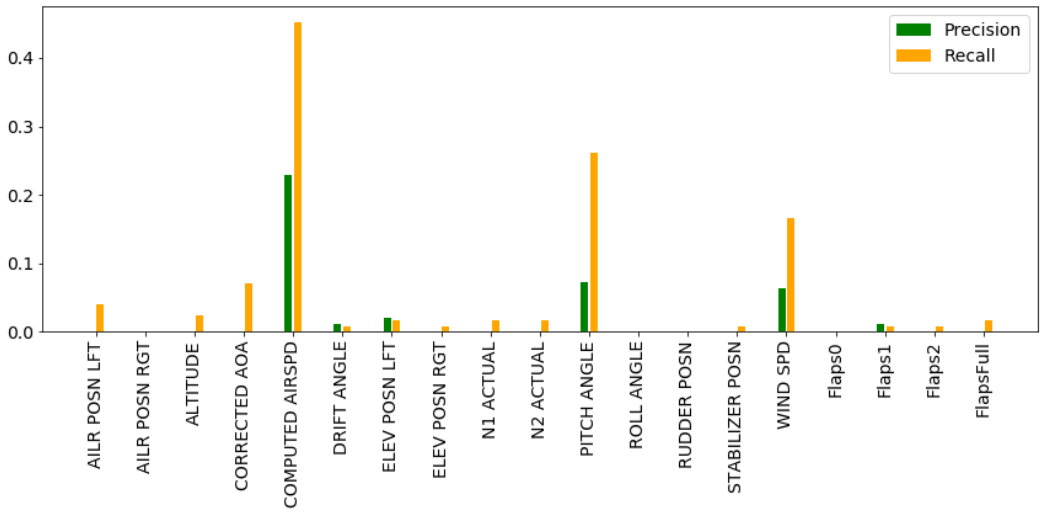

Fig. 4 Parameter importance obtained by random permutation method for classification of anomalies.

C. Multi-class Anomaly Detection during Approach to Landing

In this section we introduce a multi-class anomaly detection data set based on FOQA data from a commercial airline[‡]. These data are comprised primarily of 1 Hz recordings for each flight and similar to he data set for binary anomaly detection covering a variety of systems, including the state and orientation of the aircraft, positions and inputs of the control surfaces, engine parameters, and auto pilot modes and corresponding states. The data is 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 post-flight to flag known events and create labels. Each data instance is the 160 seconds recordings of 19 FOQA variables during the approach of the aircraft to landing (few seconds before 1000ft altitude to few seconds after 500ft altitude).

1. Training/Testing Data Setup

Using the threshold-based rules obtained by subject matter experts, we have created a multi-class anomaly detection training and testing data sets containing 18313 and 6105 samples, respectively. The data is comprised of four classes: (1) Nominal, where no known anomaly of the other three classes are present in this class ($\sim$ 66.7% of total data), (2) Speed High, where the anomalies are identified based on computed airspeed being above a threshold during approach ($\sim$ 22.9% of total data), (3) Path High, where the glide slope and path of descent for landing is flagged as being high ($\sim$ 7.2% of total data), and (4) Flaps Late Setting, where the flaps are deployed late during approach to landing ($\sim$ 3.2% of total data). These events were chosen because they are all relevant metrics used to measure unstabilized approaches. Figure 1 shows a sample of a normalized trajectories from the Nominal class. Similar to the binary example, the training data is used for training each of the models and the testing is used to estimate the unbiased performance of the model. All the results in the figures are based on the performance of the models on the testing data.

Although, we have labels available for the entire training data set, for the sake of experimentation, we evaluate the performance of the models when only a small portion of the entire training set is labelled. As a result, we divide the training set into to sets of labelled data, (X_L, y_L), and unlabeled data, X_U, where $N_L + N_U = N = 18313$. Figure 5 compares the performance of the three models' accuracy, when the size of labelled set, N_L, is equal to 100 (0.55% of training data), 200, 500, and 1000 (5.5% of training data), respectively. We also report the precision and recall values per class for each of the models in Figure 11 in the Appendix. In this experiment, the labelled set is uniformly sampled across the four classes from the pool of the training data, meaning that when the size of labelled set is 100, we have 25 data per class. The data in each class is still selected randomly and due to the nature of the randomness, we repeat the training for each model 20 times and the bars show mean $+/-$ standard deviation of the performance on the testing data based on these 20 independent training. For example, in the case of 500 labelled data (125 per class), only 2.75% of the

[‡]https://c3.nasa.gov/dashlink/projects/85/

entire training data form the labelled set, (X_L, y_L), and the other 97.25% form the unlabeled set, X_U.

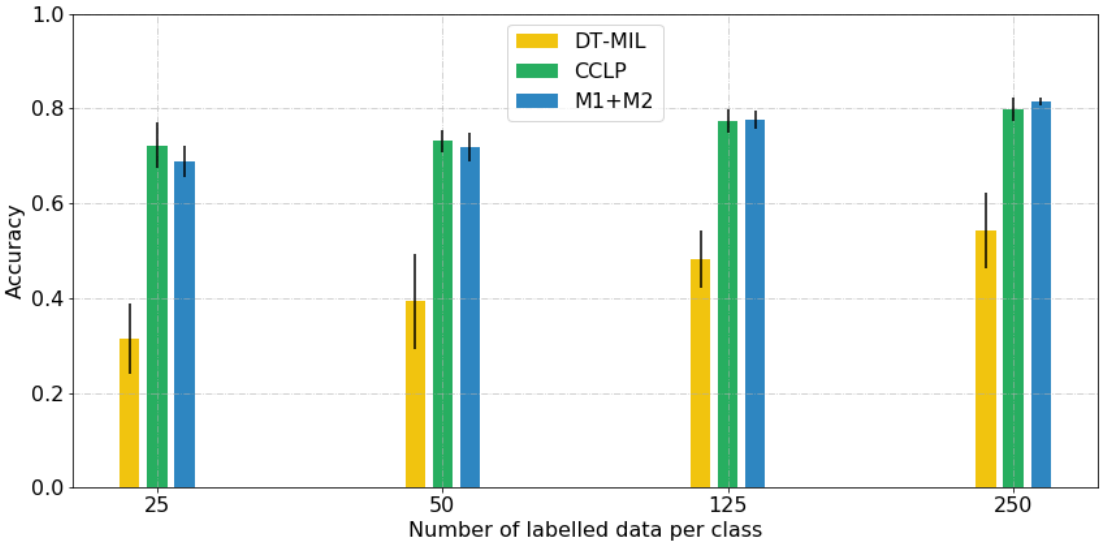

Fig. 5 This figure shows the performance of the semi-supervised models (M1+M2, and CCLP) and supervised model (DT-MIL) on the multi-class anomaly detection problem.

As it can be seen in both Figures 5 and 11, both semi-supervised models outperform the supervised model (DT-MIL) significantly and are able to achieve significantly high accuracy (72.2% for CCLP and 68.8% for M1+M2) with only 100 labelled data (25 per class), which is equivalent to 0.55% of the total data, while DT-MIL performs poorly (31.4%) with small labelled set. To better distinguish the performance of the two semi-supervised models, in Figure 6, we show the normalized confusion matrix for the case of 1000 labelled set (250 per class). Confusion matrix visualizes the proportion of the data that are correctly classified in each class, x-axis is the predicted class by the model, and y-axis is the actual class. The values that lie on the diagonal represent correct classifications and the off-diagonal values represent errors of the models or confusion of the model in detecting the data in that class. As it can be seen, M1+M2 has a better performance in distinguishing the Nominal class, while CCLP performs better in distinguishing the Anomalous classes, while sacrificing the accuracy of detecting Nominal examples.

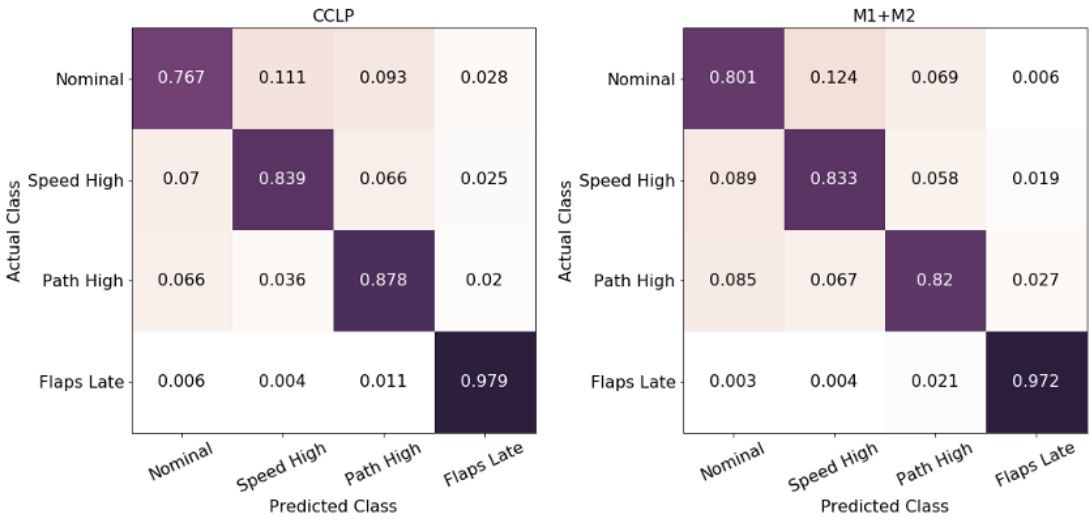

Fig. 6 Confusion matrix for semi-supervised models in the case of 1000 labelled data.

To quantify the important parameters in identifying each of the four classes, we use random permutation (similar to the binary problem), but combine precision and recall into their harmonic mean, i.e., F1-Score. In order to do this, we first calculate the precision and recall per class, where the specific class is considered the *positive* class and all other three classes are considered the *negative* class. 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 classification of each of the four classes using random permutation method. In the case of each class of anomaly we have: (1) Speed High: the top three features to identify this class are corrected angle of attack (AOA), heading of the aircraft (selected and true headings) and the 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 that is 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); and (3) Flaps Late Setting: the top three features being corrected angle of attack and computed airspeed (which both indicate information about speed of the aircraft) and the Trailing Edge (T.E.) Flap Position (which is the main feature that is used to flag these events). Figures 13, 14, and 15 show the mean +/− standard deviation of the top features in each class against the Nominal class. For example, in the case of Flaps Late class, you can see that the late deployment of the Flaps (which is the anomaly) will result in a higher than normal airspeed in approach to landing. That is 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 overall between this anomaly and Speed High events during the approach and can be investigated further in the future work.

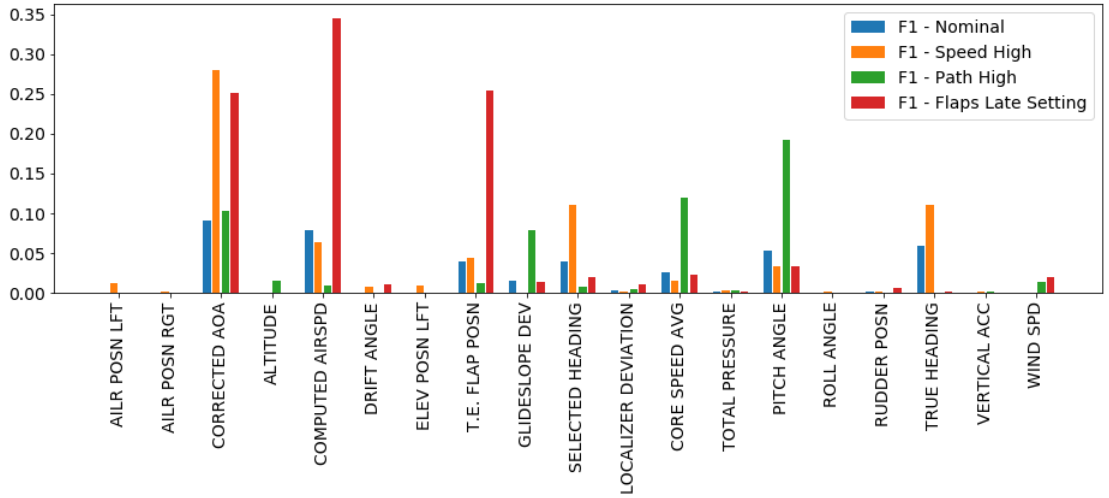

Fig. 7 Parameter importance obtained by random permutation method for classification of anomalies.

D. Latent Space Configuration

In this section, we investigate the latent space configuration of the semi-supervised models and we use the multi-class anomaly detection problem as a case study. In order to be able to visualize the 256-dimensional latent space of M1+M2 and CCLP models in 2D, we use t-Distributed Stochastic Neighbor Embedding (t-SNE) [19] implementation in `Scikit-Learn` with perplexity hyper-parameter fixed at 50. Figure 8 compares the 2D visualization of 256-dimensional latent space of CCLP and M1+M2 color-coded by the class that each data point belongs to.

As mentioned before, M1+M2 feature encoding is fully unsupervised and as expected, the clusters that are shaped 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 harder to interpret such latent space and significant data post-processing is needed to identify why and how different clusters have shaped in such latent space. On the other hand, the latent space of CCLP model shows compact clustering

of the data of different classes in the latent space. This is not a surprising outcome, since CCLP's objective function in Eq. (5) enforces compact clustering of the data belonging to the same class via label propagation technique.

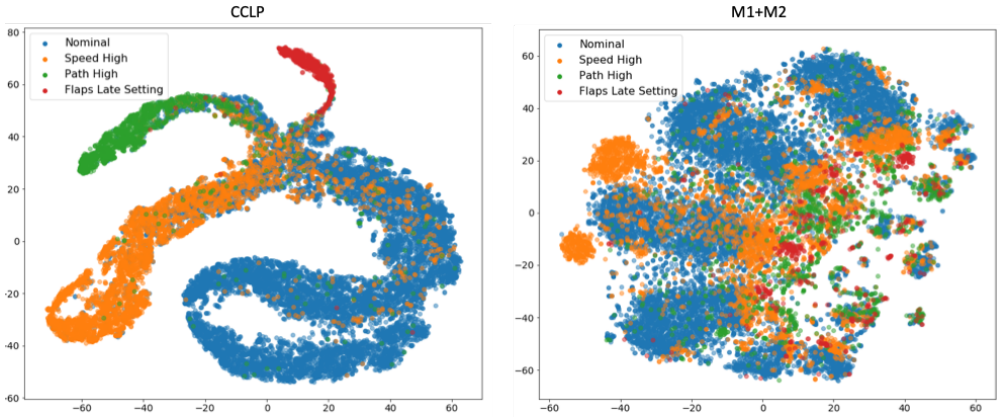

Fig. 8 2D visualization of 256-dimensional latent space of CCLP and M1+M2 using t-SNE, color-coded based on the class.

An interesting pattern that is observed in the visualization of the CCLP's latent space is that there is a central region where the four clusters (one per class) are merging into one another. We were interested to see if such cluster is actually shaping in the 256-dimensional latent space and is not an artifact of t-SNE's nonlinear mapping to 2D. In order to examine this, we applied agglomerative clustering (implemented in `Scikit-Learn`) with five (number of class plus one) clusters in the 256-dimensional latent space. Figure 9 right panel shows the 2D visualization of the clusters shaped in the latent space, where we confirmed that the fifth cluster (colored purple) is shaping exactly in the area of central region where the clusters of data belonging to each class are merging. The intensity of the colors in this graph correspond to the distance of each point to the center of the cluster that it belongs to, meaning more transparent points are further away from center of their clusters.

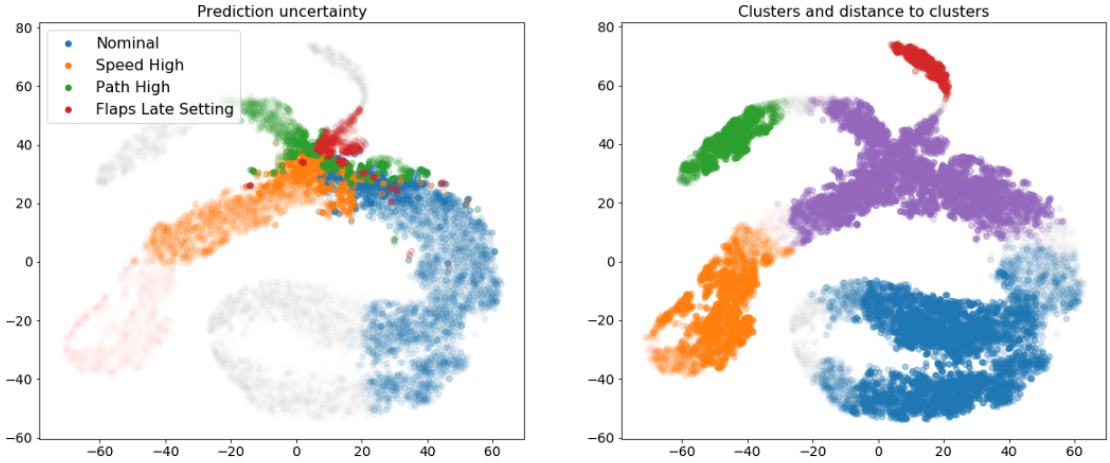

Fig. 9 This figure shows the 2D visualization of the latent space of CCLP model, where (left) emphasizes the prediction uncertainty of the model's classifier, and (right) shows five clusters shaped in the latent space and distance of the points to the center of their clusters.

After confirming that the data in the central region are being clustered together, we hypothesized that this central region is filled with data points that are hard to classify. To validate this hypothesis in Figure 9 left panel we have visualized the prediction uncertainty of the CCLP's classifier. We quantify the prediction uncertainty of the model as entropy of the output of the `softmax` layer of the classifier. The output of this layer is 4-dimensional vector, where each

element i denote the probability that the data belong to class i, for $i \in \{1, 2, 3, 4\}$. As a result, the entropy can be defined 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 data X. Points that are higher in color intensity are the ones that have higher prediction uncertainty (i.e., entropy). As it can be seen, the majority of the points that classifier has a hard time identifying their class belong to this central cluster (purple cluster in the right panel). On the other hand, data points that are easier to classify are compactly clustered away from this central cluster. This opens up new avenues and ideas for designing active learning strategies to identify the most valuable subset of unlabeled data that can be processed and studied by the subject matter experts for assigning new labels to improve the prediction performance.

VI. Conclusion

The standard practice in the aviation domain for anomaly detection is exceedance detection, which is unable to identify unknown risks and vulnerabilities. Supervised learning can overcome this challenge with the main drawback of reliance on the need for sufficient amount of processed and labelled data to reach optimum performance. However, in many real-world applications, such as aviation, these labelled data are either not available or sparse. As a result, majority of the development in the literature focus on unsupervised reasoning of the data to identify anomalies in high-dimensional time series of flight. Unfortunately, due to the nature of unsupervised learning, majority of such methods suffer from high number of false alarms and low accuracy in complex settings, which limits their applicability. In order to address the challenges of both unsupervised and supervised reasoning of aviation data, in this article we developed a robust and more explainable semi-supervised model that takes advantage of the entirety of the available data, the majority unlabeled data and the minority labelled data, and identify precursors for anomalies in aviation data with high accuracy and low number of false alarms.

We specifically deploy two successful deep semi-supervised classification models developed in the machine learning literature, named M1+M2 [14] and Compact Clustering via Label Propagation (CCLP) [15], for anomaly detection in high-dimensional and heterogeneous time series of FOQA data. In order to validate these models and benchmark their performance, we developed two case studies: (1) binary anomaly detection during take-off phase of commercial aircraft, and (2) multi-class anomaly detection during approach for landing of commercial aircraft. We compare the performance of the two semi-supervised models with the state-of-the-art supervised model [7], developed specifically for anomaly detection in aviation data.

We show in Figures 3 and 5 that both semi-supervised models significantly outperform the supervised models with availability of a minimal labelled set. For example, we see that CCLP model outperforms DT-MIL significantly with 72.2% accuracy compared to 31.4%, with availability of only 100 labelled data, which is equivalent to 0.55% of the total data. This accuracy reaches over 80% with 1000 labelled data available (5.5% of total data) for the CCLP model, while the supervised model (DT-MIL) performs below 60%.

We further investigate the explainability of the semi-supervised models and specifically, in the case of CCLP model, we show that the latent feature space of CCLP model shows compact clustering of the data of different classes (Figure 8 left panel). A very interesting finding was observed by comparing the clusters formed in the latent space against the classification's prediction uncertainty in Figure 9. We discovered that the majority of the points that the classifier has trouble predicting class labels, belong to this central cluster (purple cluster in the right panel of Figure 9). On the other hand, data points that are easier to classify are compactly clustered away from this central cluster. This opens up new avenues and ideas for identifying the most valuable subset of unlabeled data to improve the performance of the model, if new labels were to be obtained by subject matter experts.

Appendix - Additional Figures

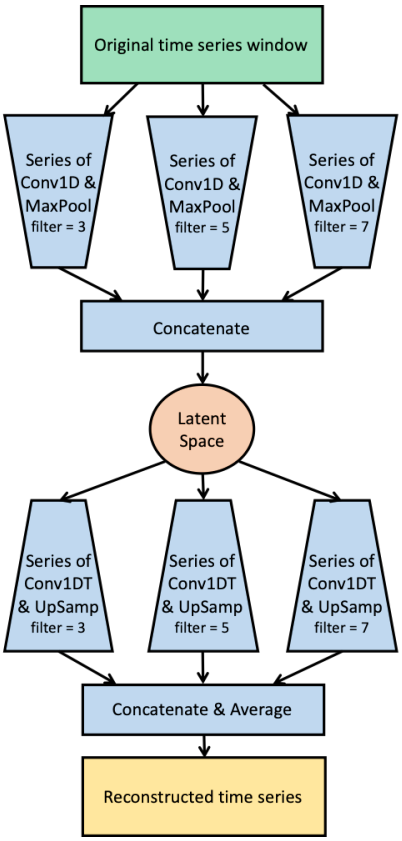

Fig. 10 Convolutional Variational Auto-Encoder architecture [12] that is used as M1 model in this article.

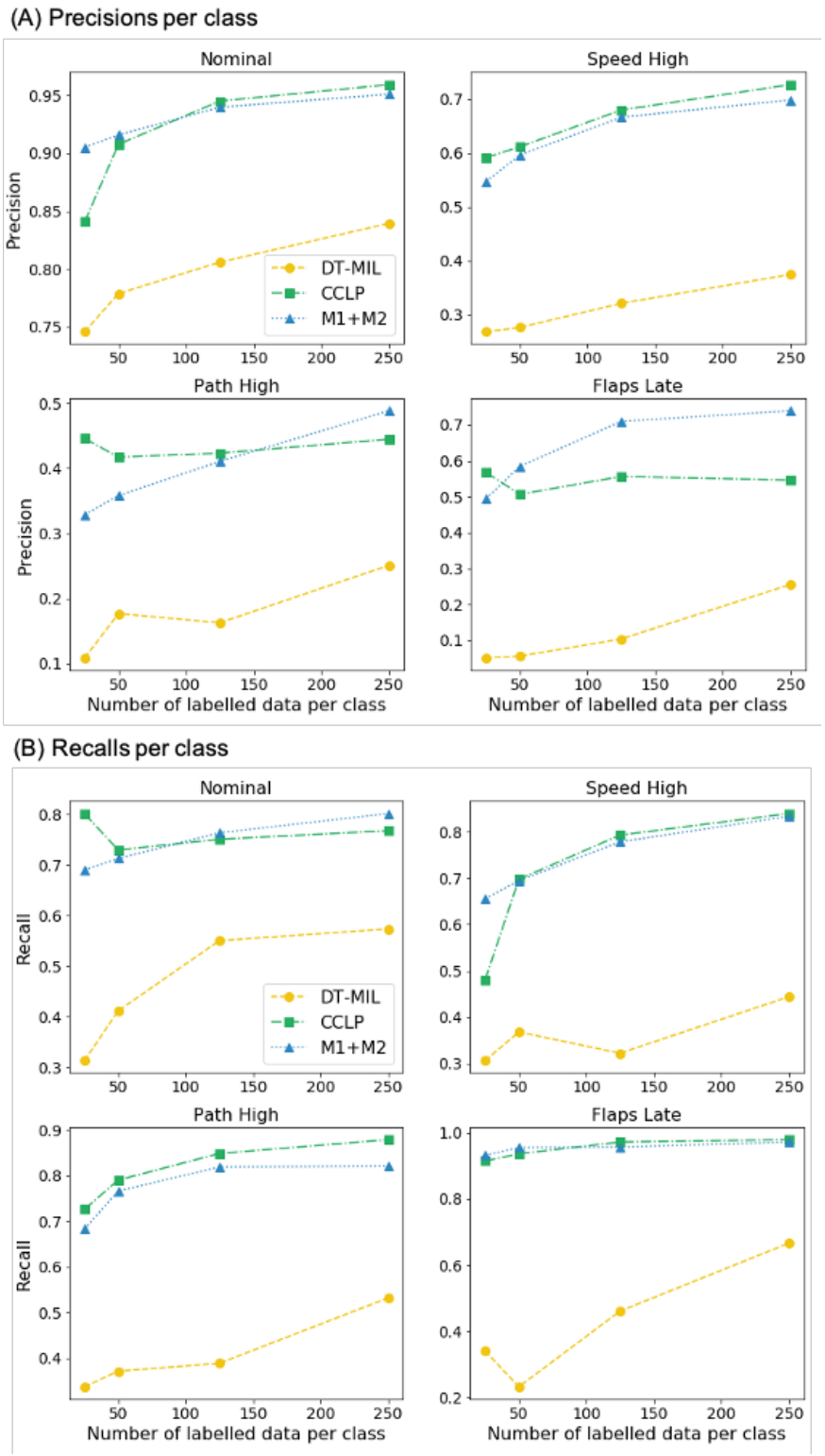

Fig. 11 Precision and recall per class for semi-supervised (CCLP and M1+M2) and supervised (DT-MIL) models in the multi-class anomaly detection problem.

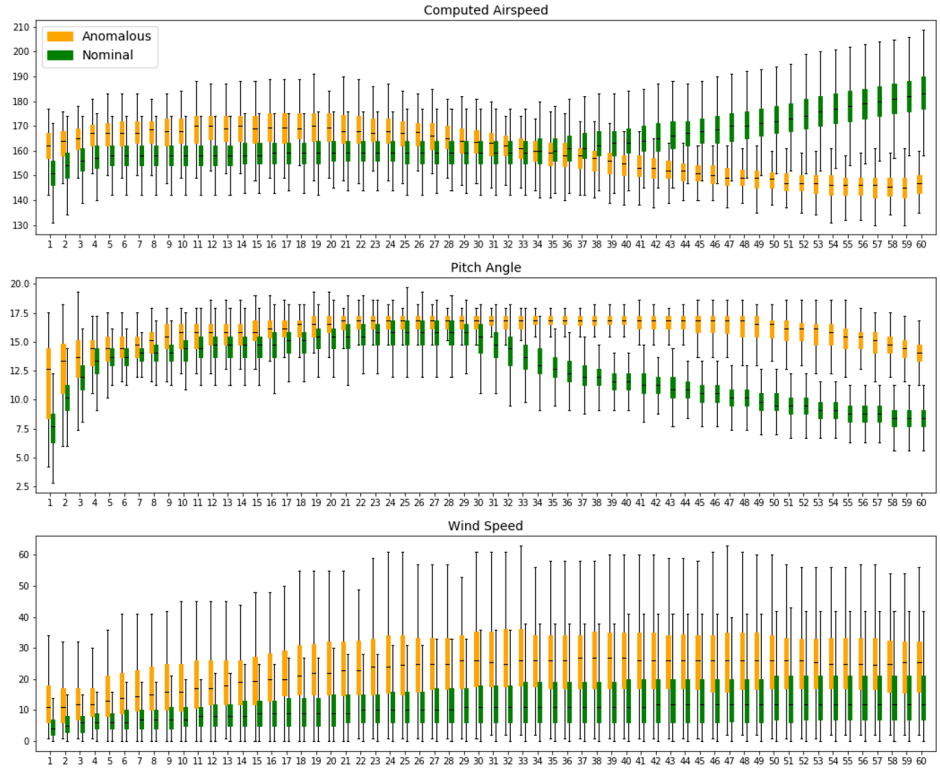

Fig. 12 Mean +/− standard deviation of trajectories of top features for take-off anomalies versus nominals.

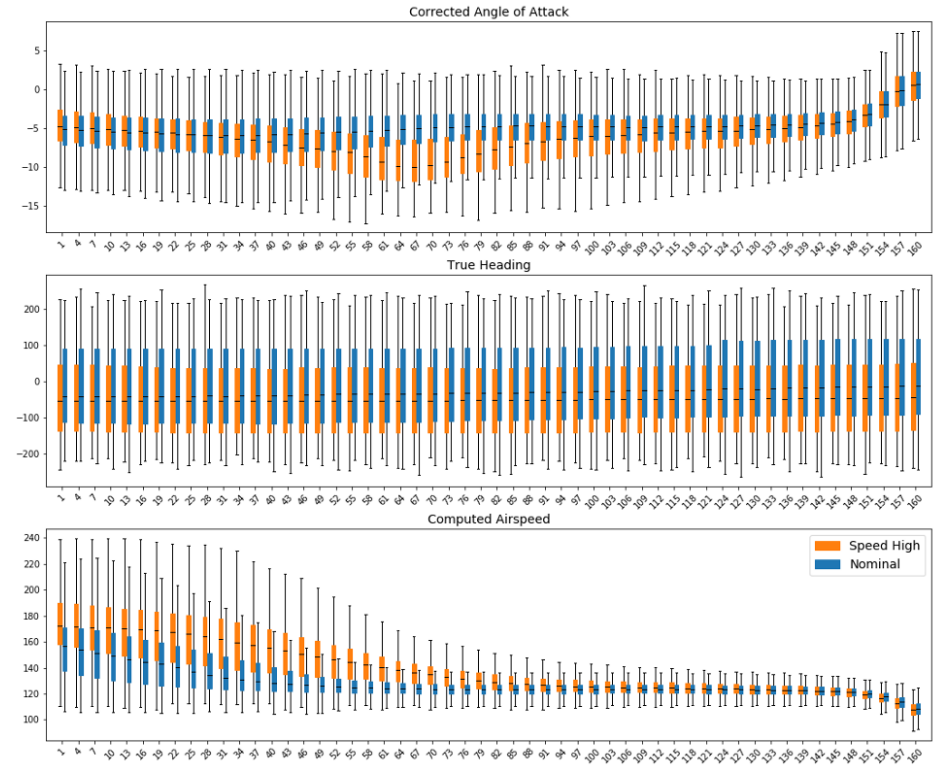

Fig. 13 Mean +/− standard deviation of trajectories of top features for speed high anomalies versus nominals.

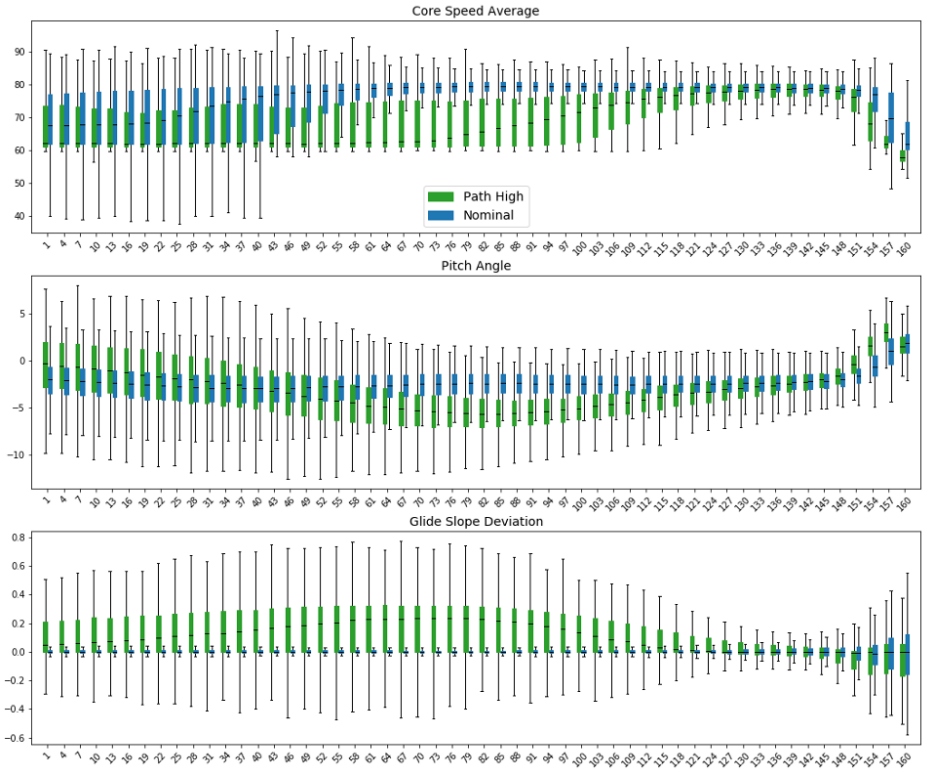

Fig. 14 Mean +/− standard deviation of trajectories of top features for path high anomalies versus nominals.

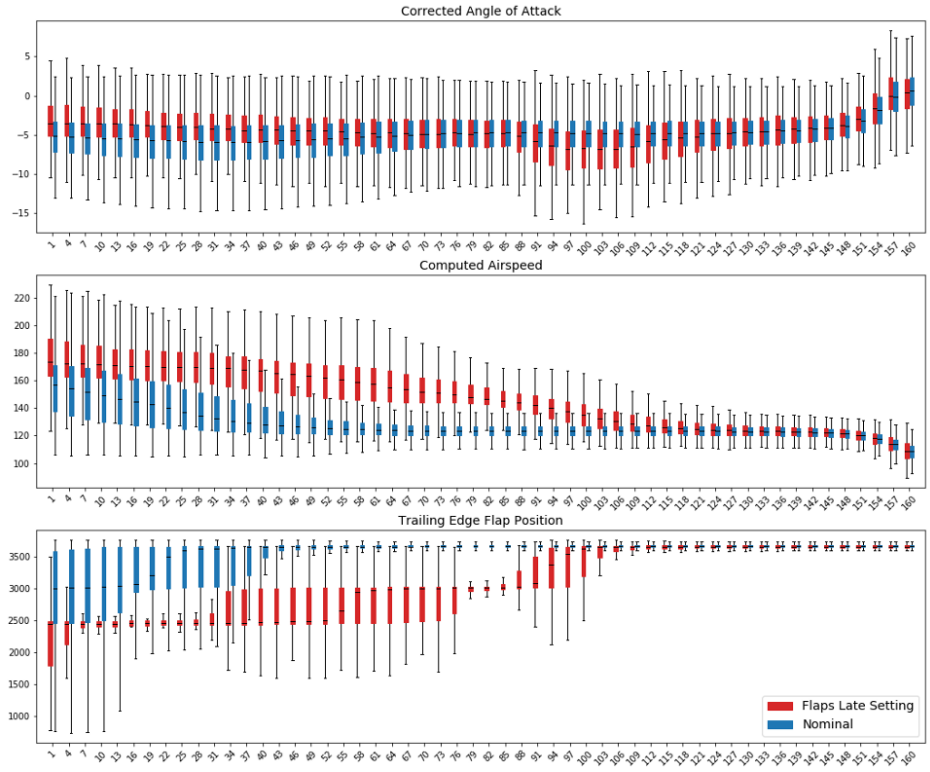

Fig. 15 Mean +/− standard deviation of trajectories of top features for flaps late anomalies versus nominals.

Acknowledgments

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

References

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

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

[3] Sprung, M. J., Chambers, M., and Smith-Pickel, S., "Transportation Statistics Annual Report 2018," , 2018. URL https://rosap.ntl.bts.gov/view/dot/37861, technical Report.

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

[5] Federal-Aviation-Administration, "Flight Operational Quality Assurance," *Technical Report*, , No. 120-82, 2004. URL 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, 2020, p. 73.

[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*, 2018, pp. 406–415.

[8] 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*, 2003, pp. 29–38. https://doi.org/https://doi.org/10.1145/956750.956758.

[9] Iverson, D. L., "Inductive System Health Monitoring," *Proceedings of the International Conference on Artificial Intelligence*, 2004.

[10] 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*, 2010, pp. 47–56.

[11] 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*, 2013, p. 4386.

[12] Memarzadeh, M., Matthews, B., and Avrekh, I., "Unsupervised Anomaly Detection in Flight Data Using Convolutional Variational Auto-Encoder," *Aerospace*, Vol. 7, 2020, p. 115.

[13] Chapelle, O., Scholkopf, B., and Zien, A., *Semi-Supervised Learning*, The MIT Press, Cambridge, MA, USA, 2006.

[14] Kingma, D., Rezende, D., Mohamed, S., and Welling, M., "Semi-supervised learning with deep generative models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. URL https://arxiv.org/abs/1406.5298.

[15] 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. URL https://arxiv.org/abs/1406.5298.

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

[17] 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.

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

[19] van der Maaten, L., and Hinton, G., "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 2579–2605.