Novel Active Learning Framework for Anomaly Detection in Aviation with Experts in the Loop

Milad Memarzadeh[1], Bryan Matthews[2], Thomas Templin[3], Aida Sharif Rohani[4], and Daniel Weckler[5]

Anomaly detection in commercial aviation is an extremely challenging yet crucial task. Accurately detecting operationally significant anomalies can save civilian lives and/or result in significant savings in maintenance cost. The current practice uses manually tuned rule-based mechanisms to flag exceedances from pre-defined safety boundaries. However, this system cannot identify unknown risks and emerging vulnerabilities. Recently, innovative approaches based on machine learning have been utilized to automate anomaly detection. However, there are limits to their applicability in the field of aviation due to several challenges: (1) Properly reviewed data is scarce in aviation and, as a result, supervised learning cannot reach optimal performance. (2) Operationally significant anomalies do not coincide with statistically significant ones and, as a result, unsupervised learning fails to provide reliable and robust performance. In this paper, we propose SALAD, a Semi-supervised Active Learning framework for Anomaly Detection, which detects operationally significant anomalies in flight operational quality assurance data. The developed framework works with vast amounts of unlabeled data as well as a small quantity of labeled data reviewed by subject matter experts to reliably identify safety anomalies in flight operations. Moreover, the model's active learning strategy allows it to detect unknown anomalies that might emerge in the system. We validate performance of SALAD with a real-world case study of anomaly detection during the approach to landing of commercial aircraft. We specifically show that the proposed framework reaches reliable performance when only one percent of the data is labeled and can identify unknown anomalies effectively.

Introduction and Related Work

Detecting operationally significant anomalies in aviation data is an extremely challenging task. Such anomalies may be precursors to adverse events and incidents, and the early detection of off-nominal patterns can save civilian lives as well costs to operate and maintain the aircraft. The automated detection of aviation anomalies is confronted with the following challenges: (1) The size of the overall data volume properly reviewed by subject matter experts (SMEs) is very small compared to the vast amount of data generated every day by the National Airspace System (NAS); (2) The exact definition of operationally significant anomalies changes from aircraft to aircraft and depends on how air traffic controllers manage the flow of aircraft, environmental conditions, the geometry of the arrival/departure airport and airspace regulations and procedures; (3) Safety reviews relying on human data inspection are very time-consuming and ineffective and can potentially miss emerging vulnerabilities and unknown anomalies. Additionally, there may be disagreements among the SMEs regarding the labeling of the same event. At present, knowledge about adverse events in the NAS comes from after-the-fact forensic investigation aimed at determining the root cause of an incident or accident, such as the manual process that the National Transportation Safety Board (NTSB) uses when investigating accidents (National-Transportation-Safety-Board 2002).

Monitoring the operation of commercial aircraft usually employs a process called exceedance detection (Federal-Aviation-Administration 2004). This technique compares flight-relevant parameters with pre-

[1] Senior Scientist, USRA, Corresponding Author: milad.memarzadeh@nasa.gov
[2] Research Engineer, KBR.
[3] Computer Engineer, NASA Ames Research Center. Senior Member, AIAA.
[4] Software Engineer, USRA
[5] Data Engineer, KBR.

defined thresholds that were defined using domain knowledge and the results of forensic reviews of similar situations. This 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. Furthermore, this technique is incapable of detecting anomalies more complex than threshold crossings that might involve off-nominal behavior of multiple correlated parameters. Machine learning (ML) approaches can automate and enhance anomaly-detection performance by identifying statistically significant off-nominal patterns that are more complex than simple exceedances. However, not all statistically significant aviation anomalies are operationally significant, and ML-based anomaly detection requires supplementation through SME review. Reviews of SMEs (i.e., data labels) can be used as feedback to improve the performance of ML models. Anomaly-detection approaches based on ML have the potential to not only automatically identify adverse events on the basis of information furnished by experts, but also to identify unknown systematic risks and anomalies and to rank data according to its information content for the anomaly-detection task. This process can eliminate the extensive search required by human-only reviews and significantly reduce experts' engagement time. Given that expert reviews are extremely costly, an effective ML approach can also result in significant savings in operational costs by focusing the SME reviews to the the critical flights that can be used to improve the performance of future ML models.

The use of ML for anomaly detection in aviation has been gaining more attention recently because of an increase in the amount of data and the concomitant need for an automated approach that relies less on human intervention. The ML approaches developed in response to this need fall into two main categories: (1) supervised learning, which produces inference using only labeled data reviewed by experts; these approaches have demonstrated impressive performance when trained on a sufficiently large number of labeled data (Janakiraman 2018; HyinKi Lee et al. 2020; Mori 2021); and (2) unsupervised learning, which does not rely on the availability of reliable data labels and uses data-driven mechanisms to identify off-nominal patterns in the data. Different techniques have been used to find such off-nominal patterns including proximity-based methods (Bay and Schwabacher 2003; Melnyk et al. 2016), clustering-based methods (Iverson 2004; Budalakoti, Srivastava, and Otey 2009), kernel-based methods (Das et al. 2010; Matthews et al. 2013; H. Lee et al. 2020), and deep learning-based methods (Hundman et al. 2018; Memarzadeh, Matthews, and Avrekh 2020). Each of these broad categories has its own advantages and disadvantages: Supervised learning performs amazingly well when a sufficient number of reviewed and labeled data is available. However, labeling aviation data requires time-consuming and costly efforts from SMEs, which does not scale for larger datasets and makes this approach largely impractical. As a result, the size of reliably labeled aviation datasets is not large enough to allow supervised models to reach optimum performance and generalize well. Moreover, supervised approaches can only identify known risks and anomalies reviewed and labeled by experts. If a system experiences an unknown vulnerability, these models usually fail to identify it, and classify it either as nominal or as belonging to a known anomaly class. On the other hand, unsupervised learning addresses the lack of labeled data and the difficulty of review by relying only on the unlabeled data for inference. However, it suffers from a high number of false alarms and low accuracy of anomaly detection, especially when the anomaly is not a point anomaly (an anomaly that occurs during one time stamp) and in scenarios of multiple concurrent anomalies. This is particularly true for complex data such as high-dimensional heterogeneous time series. This drawback is due to the fact that statistically significant anomalies are not guaranteed to always coincide with operationally significant anomalies.

The number of studies in the aviation-safety literature that fill the gap between unsupervised and supervised ML is limited. Active learning (Sharma et al. 2016; Das et al. 2017; Sahasrabhojanee et al. 2020) has been developed to tackle this problem, where different information-theoretic or uncertainty-based methods are used to identify the most informative data (among the vast pool of unlabeled data) to be reviewed and labeled by SMEs. Although these work improve the performance and efficiency of the supervised methods by incorporating smart labeling strategies, they do not tackle the shortcomings completely and still require a fully supervised training scheme.

To address the shortcomings of both supervised and unsupervised approaches and build a framework capable of identifying unknown risks and vulnerabilities, we develop SALAD, a Semi-supervised Active Learning framework for Anomaly Detection. The framework consists of two synergistic modules: (1) a learning module, which is an interpretable semi-supervised anomaly-detection model built upon recent developments in the literature (Kamnitsas et al. 2018; Memarzadeh, Matthews, and Templin 2021), and (2) a data selection module, which identifies the most informative data, relevant to the anomaly-detection task, for future SME review/labeling. We incorporate common data-selection strategies such as random selection and the information theory-based entropy concept as well as the more recently developed information-theoretic strategy of Bayesian Active Learning by Disagreement (BALD) (Houlsby et al. 2011; Gal, Islam, and Ghahramani 2017). Moreover, we develop a simple and scalable data selection strategy based on clustering, compare its performance with the above-mentioned methods, and discuss its most fitting use cases. The proposed framework (1) is generalizable and adaptive: it can work with any multivariate time-series dataset and with both unlabeled or a combination of unlabeled and labeled data; (2) can automatically identify unknown risks and vulnerabilities in the presence of SMEs in the loop; and (3) improves the explainability of the features learned from the data so that they are more understandable to humans and relevant to downstream tasks. The semi-supervised model uses graph-theoretic approaches to identify the most uncertain unlabeled data instances given the previously provided unlabeled and labeled data. Once these data instances have been identified, they can be reviewed by SMEs, so that unknown anomalies can be detected and labeled faster.

We validate the proposed framework through the detection of anomalies and risks during approach to landing of commercial aircraft using recorded flight data similar to FOQA data. This data is primarily comprised of 1-Hz recordings for each flight. These recordings cover a variety of systems including the state and orientation of the aircraft, positions of and inputs to the control surfaces, engine parameters, and auto pilot modes and corresponding states. We evaluate and quantify the effectiveness of our active-learning framework, SALAD, using several data selection strategies, as described above. We use several quantitative and qualitative performance metrics to evaluate and compare these approaches, including accuracy of anomaly detection, early detection of unknown anomalies/risks, effectiveness of the data selection strategy, and the structure of the learned feature space of the semi-supervised model.

Method

In this paper, we develop SALAD and show its applicability to the detection of known and unknown anomalies and vulnerabilities in aviation data using a case study of approach to landing of commercial aircraft. Let us imagine that the input data is grouped into two sets: the minority labeled set, (X_L, y_L), and the majority unlabeled set, X_U, where the size of the unlabeled set is significantly larger, i.e., $|X_U| \gg |X_L|$. Figure 1 graphically illustrates the SALAD framework, which includes two synergistic modules. The first module is the learning module, which receives the input data, $\{(X_L, y_L), X_U\}$, and trains a semi-supervised model. The second module is the data-selection module, which takes the trained model as input and ranks samples in the unlabeled set based on estimated information useful for anomaly detection. Lastly, depending on the time and effort expended by SMEs, the M top-ranked samples (commonly referred to as the budget in the active learning domain), outputted by the data selection module, will be reviewed and labeled by SMEs for the next round of training. In this iterative process, the size of the labeled set increases and the size of the unlabeled set shrinks accordingly. Moreover, the number of known classes, n_c, can change as a result of the detection of a previously unknown class during the review process. In the next sections, we provide technical details of each module.

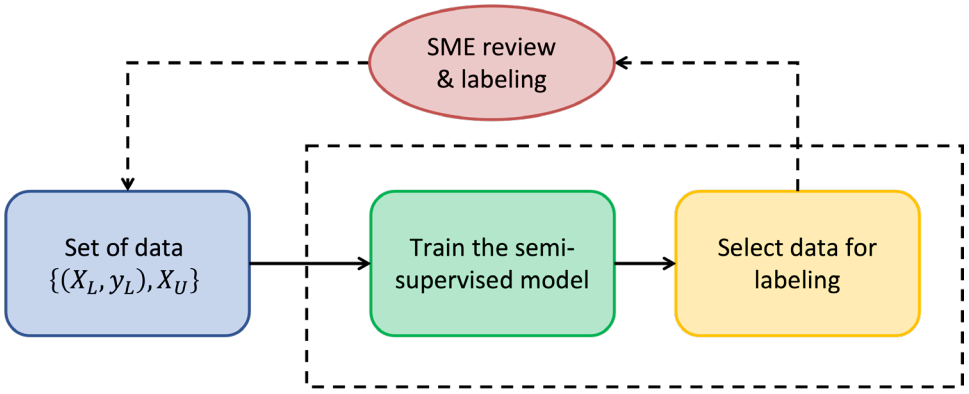

Figure 1 Graphical illustration of the SALAD framework and its synergistic components.

Learning Module: Semi-supervised Learning via CCLP

For the learning module of SALAD, we deploy Compact Clustering via Label Propagation (CCLP) (Kamnitsas et al. 2018). Our previous work on semi-supervised anomaly detection in aviation data showed the outstanding performance of the CCLP method, and therefore we select this approach for the learning module (Memarzadeh, Matthews, and Templin 2021). CCLP is a semi-supervised learning method whose main components are an encoder and a classifier as illustrated in Figure 2 The encoder, $q_\phi(z \mid x)$, is a deep convolutional neural network (its exact architecture is shown in Figure 9 in Appendix A) that maps the input data X to a feature space, Z (also referred to as the latent space). The classifier, $c_\psi(y \mid z)$, is a fully connected neural network with dropout regularization (see Figure 10 in Appendix A) that classifies the data in the feature space. Parameters ϕ and ψ represent the weights of the neural network for the encoder and the classifier, respectively.

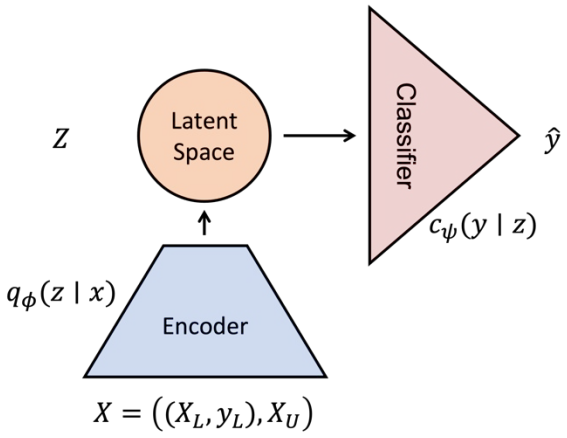

Figure 2 Graphical illustration of the learning module and CCLP's architecture.

CCLP follows the cluster assumption in semi-supervised learning and utilizes graph theory to structure the latent feature space in a way that data of the same class forms a compact cluster away from the clusters of other classes. The model is trained using all available data (labeled and unlabeled). The optimization objective of the learning module is to find a set of weights (ϕ^*, ψ^*) that minimizes the following loss function:

The first term in Eq. (1) is the classification loss and is defined as the cross entropy ($\mathcal{R}$) between the true labels for the labeled set and the predictions of the classifier. The second term is the CCLP loss, which calculates the cross entropy between the optimal transition matrix, T, and the one estimated via dynamic graph construction and label propagation, H. The variable $N = |X_L| + |X_U|$ denotes the total number of training data and S is the step of the Markov chain on the constructed graph.

$$\mathcal{L}(\phi, \psi; X) = \mathbb{E}_{(X_L, y_L)}[\mathcal{R}\left(y_L, c_\psi\left(y \mid q_\phi(z \mid X_L)\right)\right)] + \frac{1}{S}\sum_{s=1}^{S}\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} - T_{ij}\log H_{ij}^{(s)} \qquad (1)$$

To calculate H, we first obtain an adjacency matrix, A, between pairs of points in the latent feature space of the model. This matrix can be calculated using any similarity metric; in this work, we use the cosine similarity,

$$A_{ij} = \exp\left(z_i . z_j^{\mathrm{T}}\right) \tag{2}$$

where, T is the transpose operation. The estimated transition matrix H is the row-normalized version of the adjacency matrix. On the other hand, the optimal transition matrix T indicates a desired structure for the latent feature space: transition probabilities between any two data points of the same class are the same and are zero for inter-class transitions. The CCLP loss is calculated in closed form by propagating the class posteriors from the labeled set to the unlabeled set according to the random walks performed using the estimated transition matrix, H, until the equilibrium state is obtained (Kamnitsas et al. 2018).

Once the model is trained according to the loss function of Eq. (1) and the optimal weights are obtained, we switch to the data selection module to find data instances (from the pool of unlabeled data) most informative for anomaly detection and provide them to SMEs for labeling.

Data Selection Module: Baseline Strategies

As noted above, the input to the data selection module is a trained semi-supervised model and the goal is to identify the M data instances from the unlabeled set that are most relevant to the anomaly-detection task and have them labeled by SMEs. The number of data to be labeled, i.e., M, depends on the available SME time and effort. In this section we describe several metrics commonly used in the active learning literature for such a task and present a new one as well.

Random strategy

The simplest baseline method for data selection that we incorporate in our model is the random strategy. At each iteration, M data instances are randomly selected from the unlabeled set for SME labeling. This strategy is tagged as *Random* in the figures.

Uncertainty-based strategy: entropy

Uncertainty-based concepts are among the most common metrics used to identify the most informative data to be labeled in active learning. In such settings, the classifier prediction is used to identify data instances where the classifier is most uncertain to which class these data belong. The concept is based on selecting the instances that will be the most useful to the classifier by having SMEs label these points by supplying certainty. With these newly labeled points it is expected that the retrained model will have improve performance. To quantify the uncertainty of the classifier's predictions, we use the concept of entropy. As depicted in Figure 10 in Appendix A, the output of the last layer (softmax layer) of the classifier is a vector that estimates the probabilities $\hat{y}_i$ that a data instance belong to the possible classes $i \in \{1, \dots, n_c\}$. We take the entropy of that probability vector as a notion of how uncertain the classifier is about predicting each data instance; it is defined as follows,

$$\mathcal{H}(X) = -\sum_{i=1}^{n_c} \hat{y}_i \log \hat{y}_i \tag{3}$$

where, n_c is the number of known classes. In each round, the entropy of the classifier's predictions is calculated for the entire unlabeled set, and the M data instances with the highest entropy (prediction uncertainties) are assigned to SME labeling. This strategy is tagged as *Entropy* in the figures.

BALD

Bayesian Active Learning by Disagreement (BALD) is a method first proposed by Houlsby et al. (Houlsby et al. 2011) for information-theoretic active learning in classification tasks. Gal et al. (Gal, Islam, and Ghahramani 2017) extended this idea for utilization in active learning with deep neural networks. The key idea is to use a stochastic regularization technique in deep learning such as dropout as a Bayesian approximation of the model uncertainty (Gal and Ghahramani 2016). Gal et al. (Gal, Islam, and Ghahramani 2017) empirically showed that this approach is superior to other common active learning strategies such as *Random* and *Entropy* in the classification of medical imagery data, especially when the size of the labeled set is small. The BALD acquisition function selects a set of data instances that are expected to maximize the information gained about the parameters of the classifier, i.e., ψ. It is defined as,

$$\mathcal{I}(y, \psi \mid x, X_{\text{train}}) = \mathcal{H}(y \mid x, X_{\text{train}}) - \mathbb{E}_{p(\psi \mid X_{\text{train}})}[\mathcal{H}(y \mid x, \psi)] \tag{4}$$

where $\mathcal{I}$ is the mutual information, y is the predicted class, ψ denotes the classifier's parameters, $X_{\text{train}} = X_U \cup X_L$ is the data of the entire training set, $x \in X_{\text{train}}$, and $\mathcal{H}$ is the entropy (as defined in Eq. (3)). The data instances with the highest BALD acquisition-function values are those with high average prediction uncertainty, and some model parameters yield conflicting predictions with high certainty. In each round of active learning, the BALD acquisition function is the Monte Carlo estimated difference between the entropy of average class prediction probabilities based on approximate model posteriors and the average entropy of these class prediction probabilities:

$$\hat{\mathcal{I}}(y, \psi \mid x, X_{\text{train}}) = -\sum_{i=1}^{n_c} \frac{1}{K} \sum_{k=1}^{K} \hat{y}_i^k \log\left(\frac{1}{K} \sum_{k=1}^{K} \hat{y}_i^k\right) + \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{n_c} \hat{y}_i^k \log \hat{y}_i^k \tag{5}$$

where K denotes the number of Monte Carlo samples. We then select M data instances with the highest value of the approximate BALD metric for SME labeling. This strategy is tagged as *BALD* in the figures.

Results and Discussion

In this section, we first introduce a case study using a multi-class anomaly detection dataset containing time series representing commercial aircraft flight data during approach for landing. Then, we present a novel clustering-based data selection strategy. Finally, we validate and compare the performance of all strategies with two experiments: (1) a closed-set recognition experiment, where all anomaly classes are known from the first round, and (2) an experiment on the detection of unknown classes, where we quantify the performance of the framework in detecting unknown anomalies.

Multi-class Anomaly Detection during Approach to Landing of Commercial Aircraft

We created a multi-class anomaly-detection dataset based on Flight Operational Quality Assurance (FOQA) data from a commercial airline[6]. This data primarily comprises 1-Hz recordings for each flight and covers 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 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 to flag anomalous events and

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

create anomaly labels. Each data instance is a small subset of the entire flight comprising a 160 seconds-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, depending on the landing runway and airport geometry, the duration from 1000 to 500 ft altitude is less than 160 seconds. In this case, we expand the data window to include an additional period directly before reaching 1000 ft altitude.

We processed and labeled 30,522 overall data instances, which comprise four classes: (1) Nominal, where no anomaly of the other three classes is known to be present ($\sim 66.7\%$ of the total data); (2) Speed Anomaly, where the anomaly is identified based on a deviation from the target landing airspeed during approach ($\sim 22.9\%$ of the total data); (3) Path Anomaly, where the path of descent for landing is flagged as anomalous and deviating significantly from the runway's glide slope ($\sim 7.2\%$ of the total data); and (4) Control Anomaly, where the extension of flaps (control surfaces on the trailing edges of an aircraft's wing) is flagged as delayed in comparison to the expected landing configuration during approach to landing ($\sim 3.2\%$ 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.

We divide the data into sets for training/validation (80%) and testing (20%). The training/validation set is used for training and optimal selection of hyperparameters (discussed in the next section). The test set is used to report an unbiased estimate of the models' performance. All the figures presented throughout the paper are based on the results obtained by assessing the models' performance on the test set (a set unseen during training, validation, and hyperparameter tuning).

Data Selection Module: Clustering Strategy

As mentioned in the Method section, SALAD's learning module enforces a compact structure of the latent feature space, such that the data points of the same class are clustered compactly together and away from data points of other classes. Our previous endeavor to validate the semi-supervised learning module (Memarzadeh, Matthews, and Templin 2021) showed that in such a feature space, if we cluster the data into $n_c + 1$ clusters (where n_c is the number of known classes), there appears a central cluster. After studying this central cluster, we identified that this cluster is created by data points that the model is most uncertain about as to which class they belong to. We illustrate this in Figure 3 where we show the 2D visualization using t-Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton 2008) of an originally 256-dimensional latent feature space. The left panel shows the latent space color-coded by the true class that data instance belong to (blue: Nominal, orange: Speed Anomaly, green: Path Anomaly, red: Control Anomaly), the middle panel shows the results of color coding based on clustering the data in the latent space into $n_c + 1 = 5$ clusters, and the right panel illustrates the data instances that the classifier is most uncertain about. The central cluster is colored as purple in the middle panel and matches with the population of data points that the classifier is most uncertain about, shown in right panel.

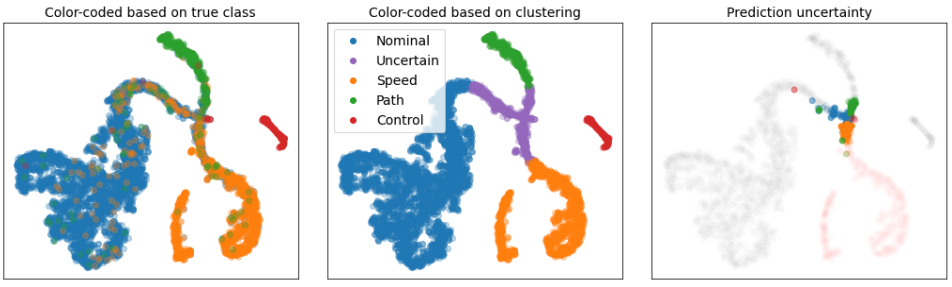

Figure 3 This figure shows a 2D visualization of the 256-dimensional latent feature space by means of t-SNE, color-coded by (left): true class, (middle): assigned cluster, and (right): prediction uncertainty.

Given a structured latent space similar to Figure 3, we applied a

simple and scalable clustering-based data selection strategy that requires a significantly less computational burden (compared to the BALD approach). This strategy, once the data instances in the latent feature space are clustered into $n_c + 1$ clusters, samples αM data instances from the uncertain cluster (the purple one in the middle panel of Figure 3), and $\frac{1-\alpha}{n_c} M$ data instances from the other, class-specific, clusters. The constant α ($0 \leq \alpha \leq 1$) represents a hyperparameter that can be set in coordination with the SMEs. For example, in the case where we have $n_c = 4$ known classes (one nominal class and three anomaly classes), if we select $\alpha = 0.2$, we are sampling new data instances for labeling uniformly from all clusters, while if we select $\alpha = 1$, we are strictly sampling all the data instances from the uncertain cluster (very similar to what the *Entropy* strategy would do). Alternatively, as we will show later, one can adopt a time-varying value of α depending on the size of labeled set, larger values initially when the labeled set is small, and smaller values later on. Our novel active-learning strategy represents an automated and systematic way of data selection for SME labeling that is more flexible and potentially performs better than either of the baseline strategies. Another advantage of this approach is that it is agnostic to the choice of clustering technique. In this article, we used *k*-means clustering with the Euclidean distance metric. However, any other clustering technique can be utilized in this framework.

Experiment I: Closed-Set Recognition

In this section, we evaluate the performance of the SALAD framework using different data selection strategies for multi-class anomaly detection in FOQA data. All three classes of anomalies are known, and the initial labeled data contains samples from all classes - a setup that categorizes the experiment as a closed-set recognition problem. The initial labeled set contains 40 samples, uniformly sampled across the four classes and randomly within each class. Then, we simulate SALAD with each data selection strategy for 20 rounds, where in each round $M = 40$ data instances from the unlabeled set are selected to be labeled by the SMEs and added to the labeled set. The *Random* strategy randomly selects M data instances in each round, *Entropy* selects them based on the uncertainty of the classifier's prediction (calculated according to Eq. (3)), *BALD* selects data based on the acquisition function defined in Eq. (4), and the *Clustering* strategy selects data based on the clusters that were formed in the latent feature space, as described in the previous section.

Figure 4 shows the average accuracy of classification in each round. Since randomness is involved in the selection of the initial set, the figures show average results across 20 independent simulations. As more data instances are labeled, the accuracy of classification consistently increases for all data selection strategies. Moreover, in the right panel we show that the impurity of the class-specific clusters formed in the latent feature space consistently improves across the different data selection strategies as more data instances are labeled. Decreasing impurity indicates that the model is forming well-separated latent space clusters for each class. We use the Gini index to measure the impurity of the clusters, as defined below:

$$\mathcal{G}_i = 1 - \sum_{j=1}^{n_c} \hat{p}_{i,j} \tag{6}$$

where $\mathcal{G}_i$ is the Gini impurity index for cluster i, and $\hat{p}_{i,j}$ is the percentage of data belonging to class j that are mapped to the cluster i. Lower values indicate greater purity. Accuracy of classification per class is also visualized in Figure 11 of Appendix B, where we show that the *Random* strategy performs better on the majority classes (Nominal and Speed Anomaly, which account for 89.6% of data), while the *Entropy*, *BALD*, and *Clustering* approaches perform better on the minority classes (Path and Control Anomalies), where the *Random* strategy suffers substantially. Moreover, the *Clustering* approach outperforms *Entropy* and *BALD* in the majority anomalous class (i.e., Speed Anomaly) and performs better than the *Random* strategy on all anomalous classes. Overall, the *Clustering* approach performs better than the baseline strategies in this experiment (Figure 4).

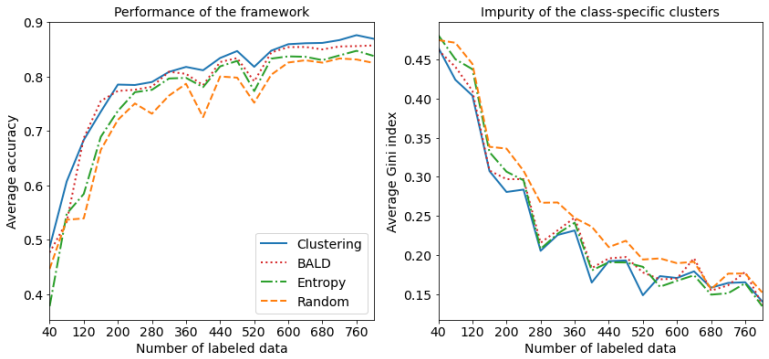

Figure 4 Average accuracy of classification as a function of the number of labeled data (left panel) and average Gini index of the class-specific clusters formed in the latent space (right panel).

It should be noted that in this approach we assign an adaptive value to hyperparameter α: We start with $\alpha = 1$ and decrease it gradually as the size of the labeled set increases until it reaches 0.2 in final rounds, at which point new data instances are sampled uniformly across the $n_c + 1 = 5$ clusters formed in the latent feature space. Figure 12 in Appendix B illustrates examples of the distributions of instances sampled for labeling across the latent feature space at different values of α. For $\alpha = 1$ all data points are sampled from the uncertain cluster (purple cluster in middle panel of Figure 3), while, as the value of α decreases, data instances are sampled more uniformly across different latent-space clusters.

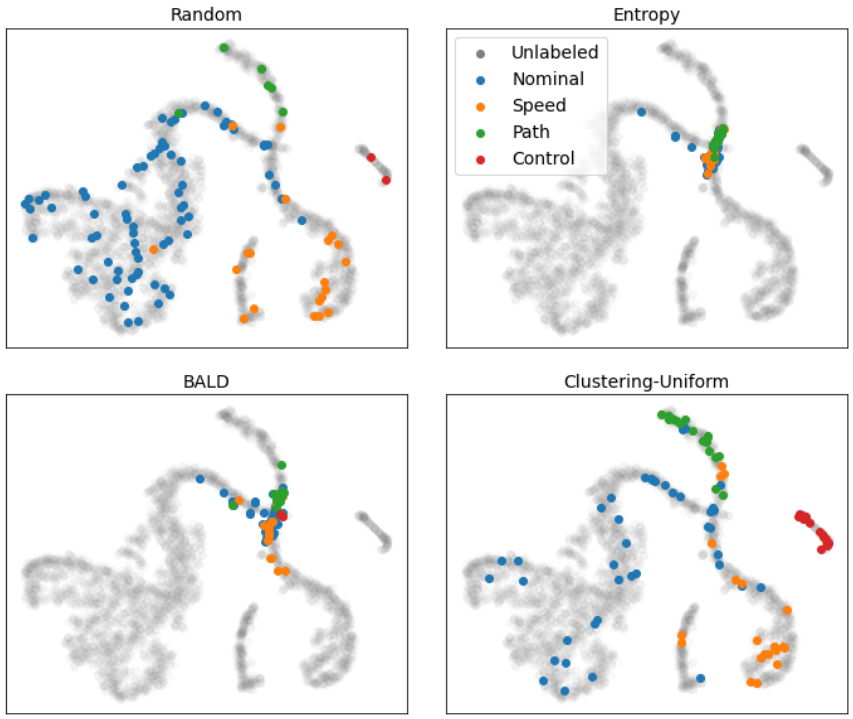

Figure 5 Distribution of data instances sampled for future labeling according to each data selection strategy. For the Clustering approach, we show here the uniform-sampling case where α=0.2.

Figure 5 illustrates an example of the sampling distributions according to each of the strategies presented here, for $M = 100$. The *Random* strategy selects data instances at random, so far more data is sampled from the majority classes (Nominal and Speed) than from the minority classes (Path and Control). On the other hand, the information theory-based methods *Entropy* and *BALD* select the majority of the data instances from the uncertain central cluster. Consequently, using an information-theoretic approach could be a good strategy when the size of the labeled set is small. Such an approach roughly corresponds to employing the *Clustering* strategy with $\alpha \approx 1$. However, a more uniform and diverse data selection strategy typically performs better at later stages of the active-learning process. Such a strategy is exemplified by using the *Clustering* strategy with α close to 0.2. In Figure 5, we show the case of uniform sampling across all clusters ($\alpha = 0.2$). Compared to the *Random* strategy, it is obvious that far more data points are sampled from the minority classes when applying the *Clustering* strategy, which is a desired outcome.

Figure 6 shows the distribution of the labeled set for the different classes. As expected, the distribution of the set under the *Random* strategy converges to the true distribution of the classes. On the other hand, the

Entropy, *BALD*, and *Clustering* strategies keep the distribution more balanced across the classes and sample more data instances from the minority classes. This is a desired outcome and shows that even if the true-class distribution of the unlabeled set is imbalanced, the proposed SALAD framework produces a more balanced labeled set over the course of the active-learning process in which additional data instances are labeled, which is important for high accuracy in anomaly detection. Another interesting observation is the shift of paradigm in data selection brought about by the *Clustering* strategy: in initial rounds, due to the starting value of $\alpha = 1$, the distribution is very similar to the *Entropy* for the minority classes. However, as the value of α decreases, the sampling strategy shifts to keep the distribution of the labeled set more uniform across the four classes. As a result, the change in α over the active-learning process favors the majority classes more in the initial stages and the minority classes later on.

Experiment II: Detection of Unknown Anomalies

In this section, we report the findings of an experiment to evaluate the performance and effectiveness of the SALAD framework in detecting unknown anomalies and vulnerabilities. For this purpose, we start the framework with a small labeled set of $M = 10$ data instances, where five are randomly selected from the Nominal class and five from the Speed Anomaly class. Accordingly, these two classes are considered to be known classes, i.e., $n_c = 2$. However, the data contains two other types of anomalies (Path and Control) which are not known to the model. In this experiment, we simulate the SALAD framework with each data selection strategy for 80 rounds of labeling, where in each round, we select $M = 10$ data instances from the unlabeled set, label them, and transfer them from the unlabeled to the labeled set according to each strategy. Our assumption here is that once an instance of an unknown anomaly class (either Path or Control) was suggested for SME review and labeled properly, the model will create a new data cluster for this class in the latent feature space. Since there is randomness in the initial selection of the labeled set and training of the model, we repeat this simulation 16 times independently and present the results averaged over these independent simulations.

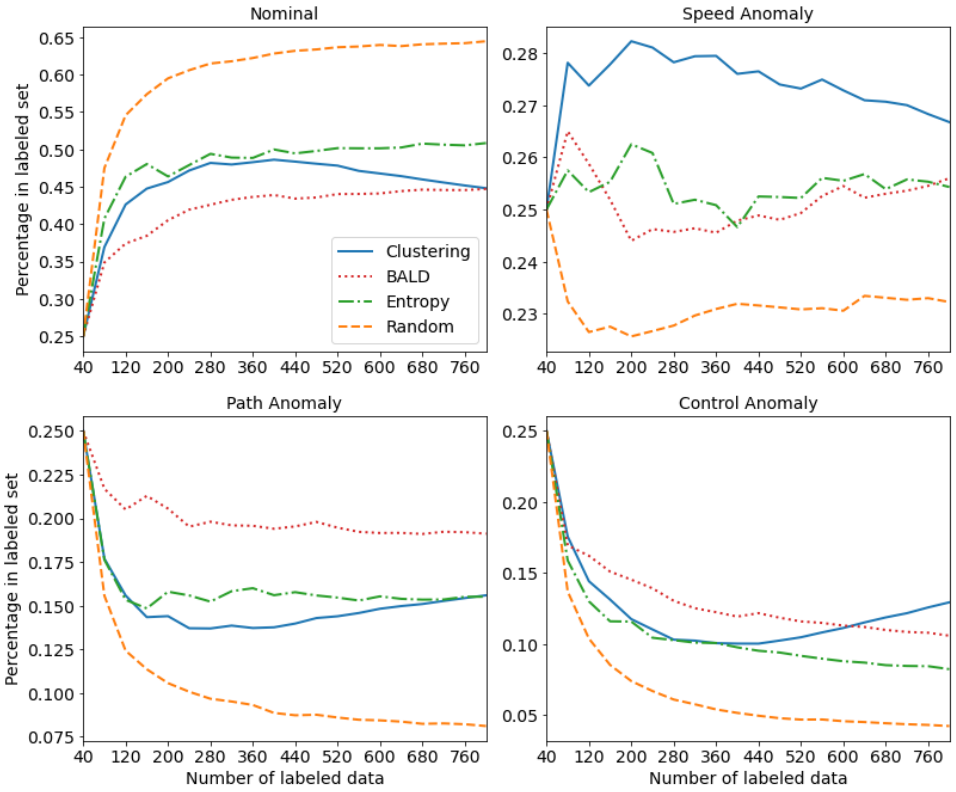

Figure 6 Class distribution of the labeled set in each round of labeling for different strategies.

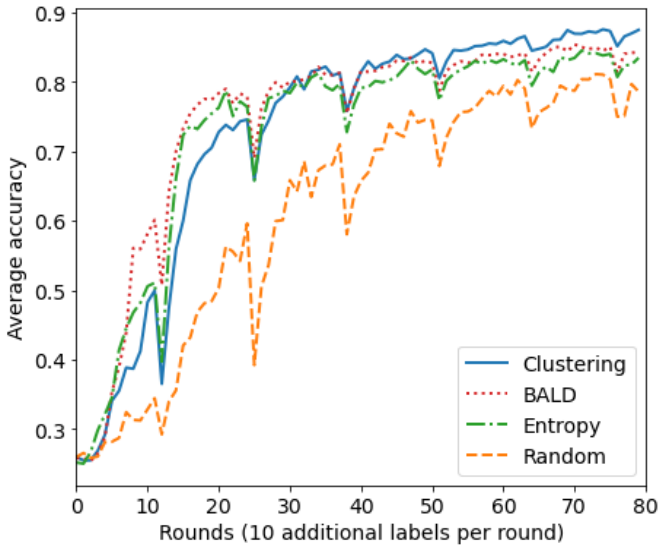

Figure 7 Average accuracy of classification for the four classes as a function of the number of labeled data.

Figure 7 shows the average accuracy of classification as a function of labeling rounds. As illustrated, the *Clustering*, *Entropy*, and *BALD* approaches significantly outperform the *Random* strategy in classification accuracy. This is due to the detection of the unknown classes in the earlier rounds of experiment. We show the per-class accuracy in Figure 13 in Appendix B, where the late detection of unknown anomalies by the *Random* strategy is evident. Another important observation is the difference between the *Clustering* strategy and *Entropy*/*BALD*. Both of these information-theoretic strategies show a greater effectiveness in detecting unknown anomalies and increasing classification accuracy for the Path and Control classes. However, the *Clustering* approach is superior to both for the Speed Anomaly class. This is due to the fact that Speed Anomaly is a majority anomaly class that is harder to distinguish from the Nominal class than the two minority anomaly classes. Figure 14 in Appendix B shows the distribution of the labeled set for the four classes over rounds of labeling, from which it is evident that the *Clustering* approach samples far more data instances from the Speed Anomaly class. Overall, *BALD* and *Entropy* perform better than *Clustering* with respect to classification accuracy in early rounds of labeling (up to round 30) and *Clustering* is superior in later rounds (after round 40).

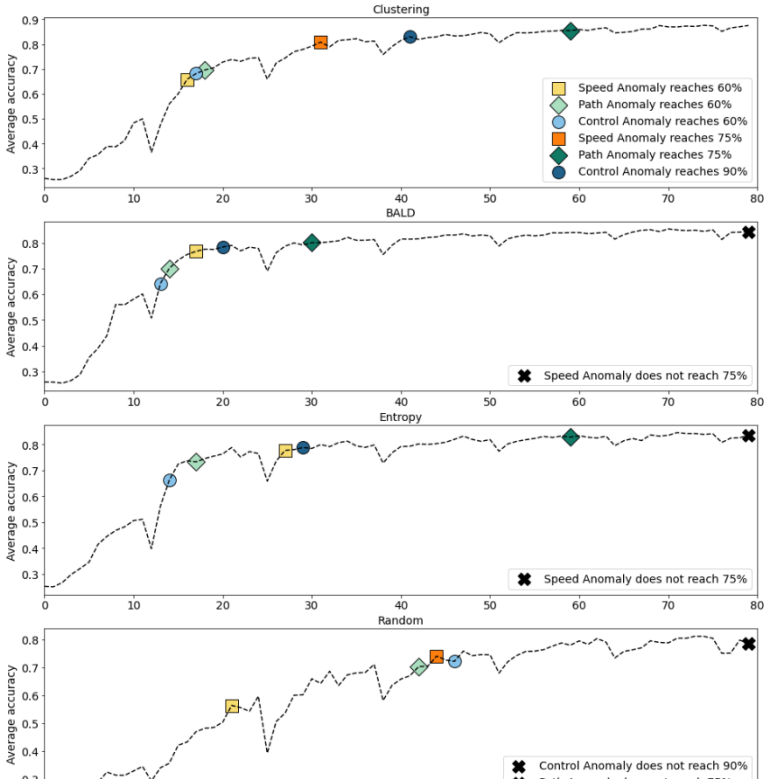

Figure 8 Effectiveness of each data selection strategy in detecting both known and unknown classes of anomaly and reaching accuracy milestones in each class.

To better understand the advantages and disadvantages of these data selection strategies, in Figure 8 we illustrate the performance and effectiveness of each data selection strategy in reaching the following milestones: (1) Each anomaly class reaches 60% accuracy as an indication of the point at which the model is first able to reliably classify anomalies; (2) The Speed, Path and Control anomaly classes reach 75%, 75%, and 90% accuracy, respectively, as an indication of the point at which the model reaches a superior detection performance for each class. These milestones were selected based on how hard it is to identify each class of anomaly (Control Anomaly is the easiest class to identify, and for this reason we

selected a harder milestone for it). As illustrated in Figure 8, the *Clustering* and *BALD* strategies reach the first set of milestones early on, in rounds 13-18 (130-180 data instances labeled), for all classes. The *Entropy* approach also reaches the first milestone for Path and Control within the same window; however, the milestone for the Speed class is only achieved in round 27. The least effective strategy is *Random*, which achieves the first set of milestones far later, namely in rounds 21, 42, and 46 for the Speed, Path, and Control classes, respectively. With regard to the second set of milestones (superior performance), *BALD* and *Entropy* do not achieve it for the Speed Anomaly class, while the *Clustering* strategy reaches it for Speed Anomaly in round 31 and *Random* in round 44. This is due to the fact that Speed Anomaly represents a majority class that is similar to the Nominal class. On the other hand, the *BALD* strategy is much more efficient for the minority anomaly classes. It reaches the superior-performance milestones for the Control and Path classes in rounds 20 and 30, respectively. This is a significant advantage of the *BALD* strategy, as the *Clustering* approach reaches the same milestones much later in rounds 41 and 59; that is, a labeled set almost double in size is required to achieve the same milestones. Lastly, we observe that the *Random* strategy does not achieve superior performance in any of the minority classes due to its imbalanced sampling scheme.

The observed differences between the information-theoretic strategies (*BALD* and *Entropy*) and the *Clustering* strategy show that none of these approaches is globally superior to another one. In some scenarios, the *Clustering* approach is more effective (Speed Anomaly), given its more balanced and diverse sampling scheme, while in other scenarios (Path and Control classes), *BALD* is more effective. Our integrated SALAD framework allows the utilization of an adaptive technique that uses *BALD* at early stages of labeling and switches to the *Clustering* strategy once the size of labeled set is large enough and the latent feature space is structured properly. Another strategy could be to rotate between *BALD* and *Clustering* in each round of labeling or use both strategies in combination to select the new subset of data instances to be reviewed and labeled by SMEs.

Conclusion

We developed SALAD, a Semi-supervised Active Learning framework for Anomaly Detection, and applied it to detect operationally significant anomalies in flight operational quality assurance data. The framework consists of two synergistic modules: a learning module and a data-selection module. The learning module is an explainable semi-supervised deep-learning model that is capable of inference using a large amount of unlabeled data and a small amount of labeled data that was reviewed by SMEs, to reliably identify safety anomalies in flights operations. The data-selection module integrates several strategies including information-theoretic approaches (i.e., *Entropy* and *BALD*) and our proposed scalable *Clustering* strategy, with the goal to identify the most informative subset of data instances from the unlabeled set to be reviewed and labeled by SMEs.

We conducted two experiments to properly validate the proposed framework. In the first experiment, we test the capability of the model in a setting where all anomaly classes are known (closed-set recognition experiment). We show in Figure 4 that on average, the *Clustering* strategy performs slightly better than the rest; however, the performance of *BALD* is competitive, followed by *Entropy* and *Random*. Specifically, the average accuracy of anomaly detection with the *Clustering* strategy reaches 75% with only 0.8% of the data (200 data instances) labeled and surpasses 85% with only 2.4% of the data (600 data instances) labeled. This is an important accomplishment, considering the scarcity of labeled data in aviation. We also emphasize in Figure 5 and Figure 6 that the *Clustering* strategy maintains a more balanced labeled set across different classes than other strategies.

In the second experiment, which studies the detection of unknown anomalies, we conduct a simulation in which the model starts with only the Nominal data and one type of anomaly as the known classes, while

the other two classes of anomalies are completely unknown to the model (open-set recognition experiment). In this experiment, our goal is to evaluate the effectiveness of the proposed framework to reliably identify both known and unknown classes of anomalies through selecting a subset of data for SME review and labeling. In Figure 7 and Figure 8 we show that information-theoretic strategies (i.e., *Entropy* and *BALD*) are faster in detecting the unknown classes of anomalies (i.e., Path and Control), but struggle to reach a high accuracy for the majority anomaly class (i.e., Speed). The *Clustering* strategy, on the other hand, is able to reach a high accuracy in classifying all anomaly classes and eventually displays a better performance than the information-theoretic and *Random* baselines. The *Random* strategy under-performs significantly in this experiment due to its inability to maintain a balanced labeled set across different classes and is not able to reach optimum performance in detecting the unknown anomaly classes even after 80 rounds of labeling.

Our future research will involve developing or adopting additional data-selection strategies that demonstrate superior anomaly-detection and classification performance. Moreover, we intend to test the SALAD framework on datasets containing additional classes of anomalies, to improve the scalability and generalizability of the model, and to eventually validate it in an operational setting.

Appendix A - Model Architecture

Figure 9 shows the architecture of the encoder. The input data passes through three parallel branches of 1D convolution operations with different number of filters (first numeric) and kernel sizes (second numeric) followed by batch normalization (BN) and ReLU activation and finally max pooling with size 2. The decoder is basically identical to a reverse encoder except that 1D transpose convolutions replace 1D convolutions, the filter sizes are in opposite order, and upsampling replaces max pooling.

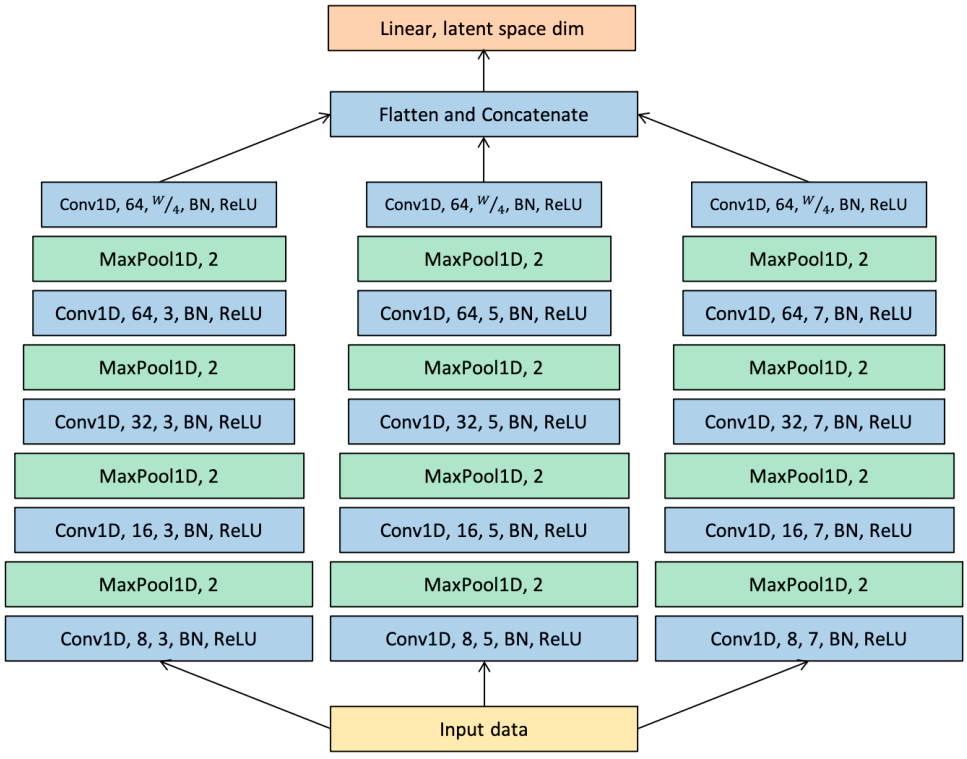

Figure 9 Exact architecture of the encoder.

Figure 10 shows the architecture of the classifier. It consists of two fully connected layers with 100 neurons each and ReLU activation functions and 50% dropout in between. The output of the second layer enters a linear layer with Softmax activation and n_c number of neurons, where n_c is the number of classes in the training data.

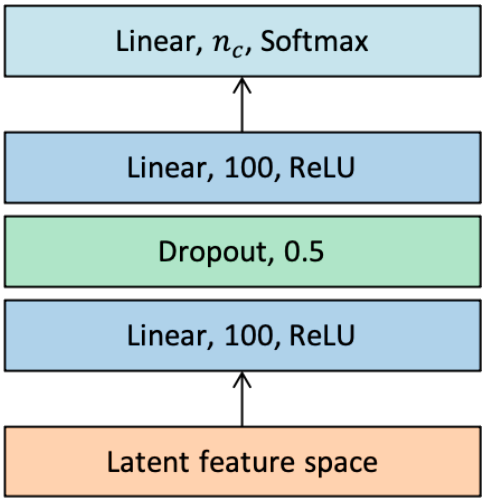

Figure 10 Exact architecture of the classifier.

Appendix B - Additional Figures

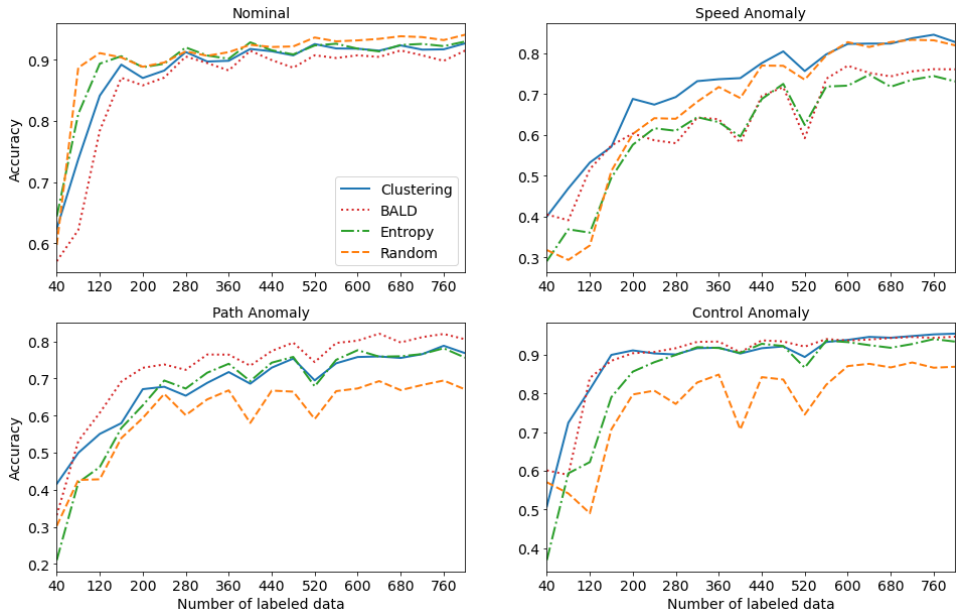

Figure 11 Per-class accuracy of classification in the closed-set recognition experiment as a function of the number of labeled data in the entire training set for each data selection strategy.

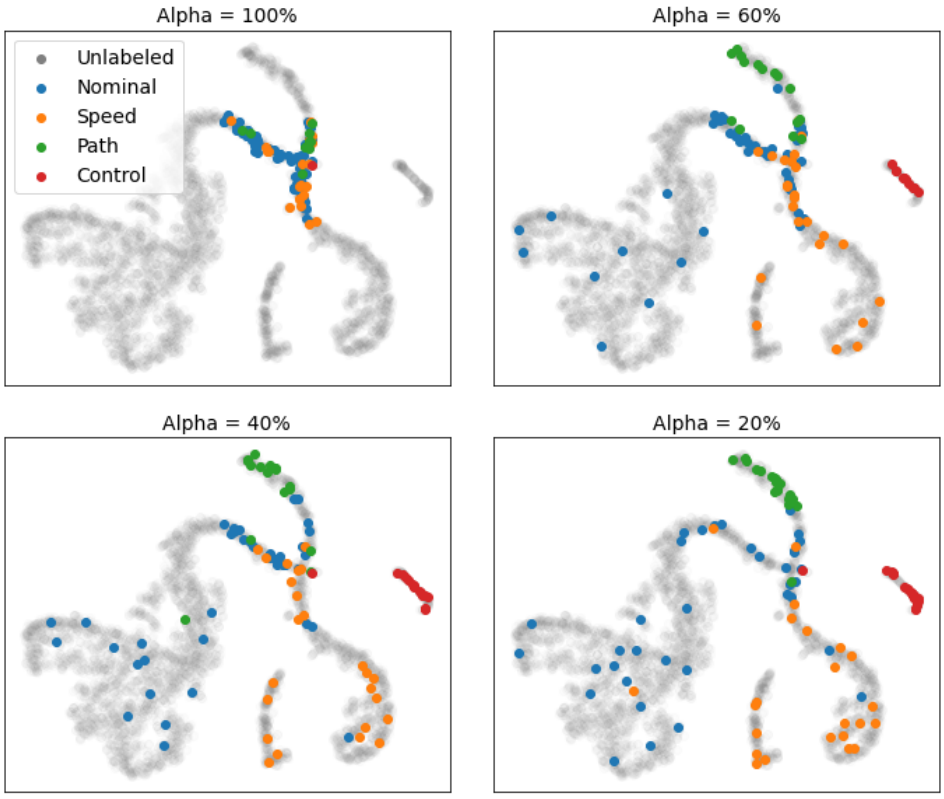

Figure 12 Distribution of data instances sampled for SME labeling based on the Clustering approach with different values of the hyperparameters α.

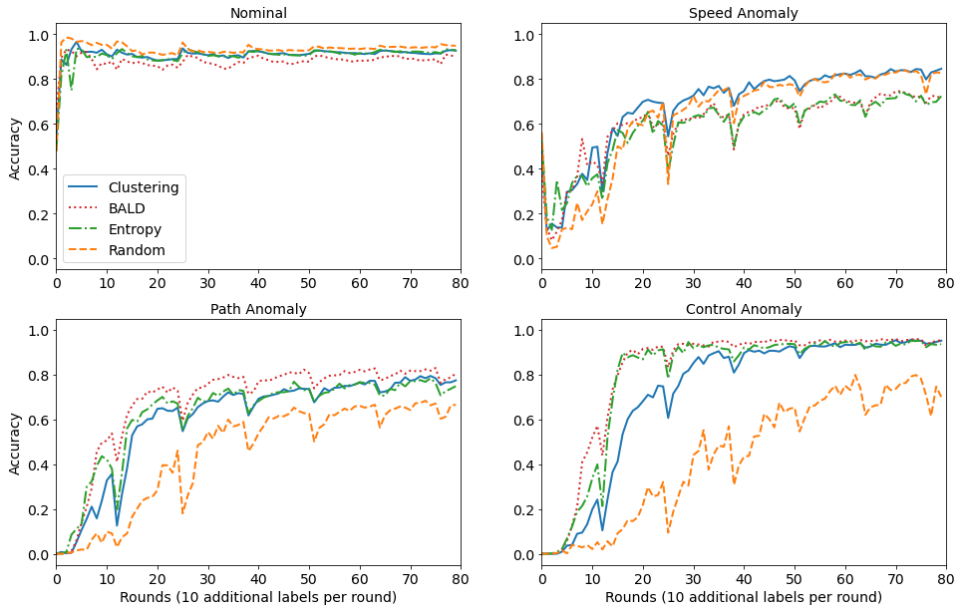

Figure 13 Per-class accuracy of classification as a function of the number of labeled data according to each data selection strategy in the unknown detection experiment.

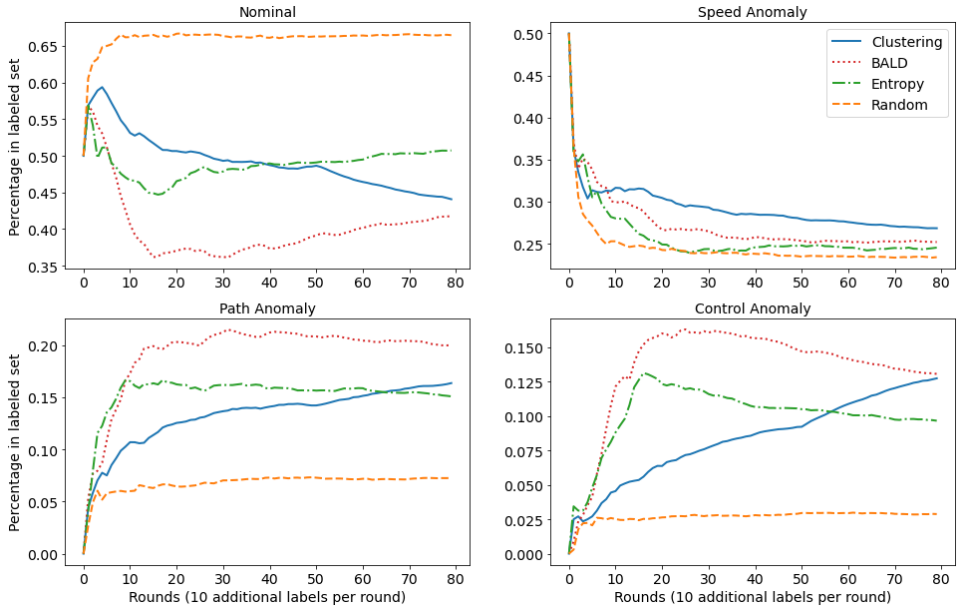

Figure 14 Class distribution of the labeled set over rounds of labeling for different strategies in the experiment on the detection of unknown anomalies.

Funding Sources

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

References

[1] Bay, Stephen D., and Mark Schwabacher. 2003. "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*, 29–38. doi:10.1145/956750.956758.

[2] Budalakoti, Suratna, Ashok N. Srivastava, and Matthew E. Otey. 2009. "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* 39 (1): 101–13. doi:10.1109/TSMCC.2008.2007248.

[3] Das, K., I. Avrekh, B. Matthews, M. Sharma, and N. Oza. 2017. "Ask-the-Expert: Active Learning Based Knowledge Discovery Using the Expert." *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD)*. Springer International Publishing, 395–99. doi:10.1007/978-3-319-71273-4_38.

[4] Das, Santanu, Bryan Matthews, Ashok N. Srivastava, and Nikunj Oza. 2010. "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*, 47–56. doi:10.1145/1835804.1835813.

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

[6] Gal, Yarin, and Zoubin Ghahramani. 2016. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.” *In Proceedings of 33rd International Conference on Machine Learning (ICML), New Tork, NY, USA* 48: 1050–9. doi:10.5555/3045390.3045502.

[7] Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. 2017. “Deep Bayesian Active Learning with Image Data.” *In Proceedings of 34th International Conference on Machine Learning (ICML)* 70: 1183–92. https://arxiv.org/abs/1703.02910.

[8] Houlsby, Neil, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. “Bayesian Active Learning for Classification and Preference Learning.” https://arxiv.org/abs/1112.5745.

[9] Hundman, Kyle, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. “Detecting Spacecraft Anomalies Using Lstms and Nonparametric Dynamic Thresholding.” *In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–95. doi:10.1145/3219819.3219845.

[10] Iverson, David L. 2004. “Inductive System Health Monitoring.” *Proceedings of the International Conference on Artificial Intelligence*. https://ti.arc.nasa.gov/m/groups/intelligent-data-understanding/ICAI2004-Iverson.pdf.

[11] Janakiraman, Vijay Manikandan. 2018. “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*, 406–15. doi:10.1145/3219819.3219871.

[12] Kamnitsas, K., D.C. Castro, L. Le-Folgoc, I. Walker, R. Tanno, D. Rueckert, B. Glocker, A. Criminisi, and A. Nori. 2018. “Semi-Supervised Learning via Compact Latent Space Clustering.” *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2459–68. https://arxiv.org/abs/1406.5298.

[13] Lee, H., G. Li, A. Rai, and A. Chattopadhyay. 2020. “Real-Time Anomaly Detection Framework Using a Support Vector Regression for the Safety Monitoring of Commercial Aircraft.” *Advanced Engineering Informatics* 44: 101071. doi:10.1016/j.aei.2020.101071.

[14] Lee, HyinKi, Sasha Madar, Santusht Sairam, Tejas G. Puranik, Alexia P. Payan, Michelle Kirby, Olivia J. Pinon, and Dimitri N. Mavris. 2020. “Critical Parameter Identification for Safety Events in Commercial Aviation Using Machine Learning.” *Aerospace* 7: 73. doi:10.3390/aerospace7060073.

[15] Maaten, L. van der, and G. Hinton. 2008. “Visualizing Data Using T-Sne.” *Journal of Machine Learning Research* 9: 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html.

[16] Matthews, Bryan, Ashok N. Srivastava, John Schade, Dave Schleicher, Kennis Chan, Richard Gutterud, and Mike Kiniry. 2013. “Discovery of Abnormal Flight Patterns in Flight Track Data.” *Proceedings of 2013 Aviation Technology, Integration, and Operations Conference*, 4386. doi:10.2514/6.2013-4386.

[17] Melnyk, I., B. Matthews, H. Valizadegan, A. Banerjee, and N. Oza. 2016. “Vector Autoregressive Model-Based Anomaly Detection in Aviation Systems.” *Journal of Aerospace Information Systems* 13: 161–73. doi:10.2514/1.I010394.

[18] Memarzadeh, Milad, Bryan Matthews, and Ilya Avrekh. 2020. “Unsupervised Anomaly Detection in Flight Data Using Convolutional Variational Auto-Encoder.” *Aerospace* 7 (8): 115. doi:10.3390/aerospace7080115.

[19] Memarzadeh, Milad, Bryan Matthews, and Thomas Templin. 2021. “Multi-Class Anomaly Detection in Flight Data Using Semi-Supervised Explainable Deep Learning Model.” *Journal of Aerospace Information Systems* In Press. doi:10.2514/1.I010959.

[20] Mori, R. 2021. “Anomaly Detection and Cause Analysis During Landing Approach Using Recurrent Neural Network.” *Journal of Aerospace Information Systems*. doi:10.2514/1.I010941.

[21] National-Transportation-Safety-Board. 2002. “National Transportation Safety Board Aviation Investigation Manual Major Team Investigations.” https://www.ntsb.gov/investigations/process/Documents/MajorInvestigationsManual.pdf.

[22] Sahasrabhojanee, A., D.L. Iverson, S.R. Wolfe, K.M. Bradner, and N.C. Oza. 2020. “Active Learning Strategies to Reduce Anomaly Detection False Alarm Rates.” *Proceedings of the 37th International Conference on Machine Learning* PMLR108, Vienna, Austria. https://3fdd288d-1fa7-4a11-ba61-585aa5221507.filesusr.com/ugd/4ce291_b6bf92489b104d26bf397122745a0d12.pdf.

[23] Sharma, Manali, Kamalika Das, Mustafa Bilgic, Bryan Matthews, David Nielsen, and Nikunj Oza. 2016. “Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation.” In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 209–25. Springer. https://www.csee.umbc.edu/~kdas1/papers/ECML_activelearning.pdf.