Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation

Manali Sharma[1](✉), Kamalika Das[2], Mustafa Bilgic[1], Bryan Matthews[3], David Nielsen[4], and Nikunj Oza[5]

[1] Illinois Institute of Technology, Chicago, USA
msharm11@hawk.iit.edu, mbilgic@iit.edu
[2] UARC, NASA Ames, Moffett Field, CA, USA
[3] SGT, Inc., NASA Ames, Moffett Field, CA, USA
[4] MORi Associates, NASA Ames, Moffett Field, CA, USA
[5] NASA Ames, Moffett Field, CA, USA
{kamalika.das, bryan.l.matthews, david.l.nielsen, nikunj.c.oza}@nasa.gov

Abstract. A major focus of the commercial aviation community is discovery of unknown safety events in flight operations data. Data-driven unsupervised anomaly detection methods are better at capturing unknown safety events compared to rule-based methods which only look for known violations. However, not all statistical anomalies that are discovered by these unsupervised anomaly detection methods are operationally significant (e.g., represent a safety concern). Subject Matter Experts (SMEs) have to spend significant time reviewing these statistical anomalies individually to identify a few operationally significant ones. In this paper we propose an active learning algorithm that incorporates SME feedback in the form of rationales to build a classifier that can distinguish between uninteresting and operationally significant anomalies. Experimental evaluation on real aviation data shows that our approach improves detection of operationally significant events by as much as 75% compared to the state-of-the-art. The learnt classifier also generalizes well to additional validation data sets.

1 Introduction

As new technologies are developed to handle complexities of the Next Generation Air Transportation System (NextGen), it is increasingly important to address both current and future safety concerns along with the operational, environmental, and efficiency issues within the National Airspace System (NAS). NASA, in partnership with the Federal Aviation Administration (FAA) and industry is continuing to develop new technologies to identify previously undiscovered safety events through data mining of large heterogeneous aviation data sets that are collected on a regular basis. These techniques have the potential to discover new safety risks in the existing system or risks that did not exist previously but are a result of the implementation of the NextGen concepts. Combined with more traditional monitoring of safety, the Aviation Safety program at NASA has invested significant resources for development and use of data mining methods for identification of unknown safety and other events in Flight Operations Quality Assurance (FOQA) data [6].

Several unsupervised anomaly detection methods have been developed to identify anomalies in commercial flight-recorded data. In the absence of knowledge regarding the types of safety events that are present in the data, and absence of labels, unsupervised techniques are the only ones that have the unique ability to find previously unknown anomalies; however, they do so only in the statistical sense—the anomalies found are not always operationally significant (e.g., represent a safety concern). After an algorithm produces a list of statistical anomalies, a Subject Matter Expert (SME) must go through that list to identify those that are operationally relevant for further investigation. A very small fraction of statistical anomalies (less than 1%) turns out to be operationally relevant, so substantial time and effort is spent by SMEs in examining anomalies that are not of interest.

The goal of this work is to semi-automate the process of distinguishing between operationally significant anomalies and uninteresting statistical anomalies through use of supervised learning approaches, which require labeled instances. We propose to use active learning for training a classifier, so that SME time and effort is spent on only the most informative and critical anomaly instances. In this process, first an unsupervised anomaly detection algorithm is run on all the flight data to generate a ranked list of statistically significant anomalies. A very small percentage of these are presented to SMEs to bootstrap the active learning process. The SME provides labels for each of these instances along with an explanation about the label. A positive label indicates an operationally significant safety event whereas a negative label indicates otherwise. Based on these few labels we build an active learning system that (i) utilizes the SME's time in the most effective manner by iteratively asking for labels for few informative instances, (ii) elicits rationales/explanations from the SME for why s/he assigns a certain label to an instance, and (iii) constructs new features, based on rationales, that are incorporated in future iterations of active learning and classifier training.

Active learning for anomaly detection has been studied in the past with the goal of finding *useful* anomalies as opposed to statistical anomalies [7] where a priori knowledge of the number of rare event classes is assumed. In our application the number of types of anomalies encountered is unknown and therefore, the assumption does not hold true. Recent work in active learning has focused on eliciting richer feedback from the experts in addition to labels, to speed up the annotation process. For example, experts are asked to annotate features as relevant/irrelevant for a specific task [1], [15]. Similarly, several researchers have investigated eliciting rationales, which often correspond to highlighting a piece of text in text classification or highlighting feature values in feature-valued representations, and incorporated them into the training of classifier [14], [18]. In this work, we build on the rationale framework by allowing the domain experts to provide rationales for their classification. The main difference between our work and existing work is that in this paper we enrich the representation by creating additional features that are combinations of existing features rather than focusing on feature value distribution.

The advantages of this method are twofold: (i) it dramatically minimizes the time an SME needs to spend to find operationally significant anomalies from the long list of statistical anomalies output by any unsupervised anomaly detection method, and (ii) at the end of training, we have a classifier that can be run on the original flight operations data set to uncover many more operationally significant safety events that might have been missed in the original anomaly detection process due to the presence of overwhelming number of statistically significant, but uninteresting, anomalies. Our experiments with real aviation data show that using active learning with rationales improves $precision@5$ (defined as number of positive instances in top 5 instances ranked according to their distance from the decision boundary) results by as much as 75% compared to the state-of-the-art.

The rest of the paper is organized as follows. Section 2 discusses the data setup and the existing unsupervised anomaly detection framework. Section 3 discusses our proposed active learning algorithm and its performance is analyzed in Section 4. Section 5 discusses deployment plans. Section 6 concludes the paper.

2 Background

In this section we describe the state-of-the-art unsupervised anomaly detection method used for identifying statistical anomalies in flight operations data, followed by description of the data used in this study.

2.1 Multiple Kernel Anomaly Detection

The unsupervised anomaly detection algorithm that is currently used in the aviation safety community most frequently is Multiple Kernel Anomaly Detection (MKAD)[6] [5]. The MKAD algorithm is designed to run on heterogeneous data sets consisting of multiple attribute types including discrete and continuous. MKAD is a “multiple kernel” [2] based approach where the major advantage is the method’s ability to combine information from multiple heterogeneous data sources. The heart of MKAD is a one-class SVM model that constructs an optimal hyperplane in the high dimensional feature space to separate the abnormal (or unseen) patterns from the normal (or frequently seen) ones. This is done by solving the following optimization problem [10]:

$$\begin{array}{ll} \min & Q = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ \text{subject to} & 0 \leq \alpha_i \leq \frac{1}{\ell\nu}, \sum_i \alpha_i = 1, \rho \geq 0, \quad \nu \in [0,1] \end{array} \tag{1}$$

where α_i’s are Lagrange multipliers, ℓ is the number of data tuples in the training set, ν is a user-specified parameter that defines the upper bound on the

[6] http://ti.arc.nasa.gov/opensource/projects/mkad/

training error, and also the lower bound on the fraction of training points that are support vectors, ρ is a bias term, and K is the kernel matrix. Once this optimization problem is solved, at least $\nu\ell$ training points with non-zero Lagrangian multipliers (α) are obtained and the points for which $\{\mathbf{x}_i : i \in [\ell], \alpha_i > 0\}$ are called the support vectors. The decision function is:

$$f(\mathbf{z}) = sign\left(\sum_i \alpha_i \sum_p \eta_p K_p(\mathbf{x}_i, \mathbf{z}) - \rho\right)$$

which predicts positive or negative label for a given test vector $\mathbf{z}$. Instances with negative labels are categorized as outliers.

The classifier that we learn using active learning for differentiating between operationally significant and uninteresting anomalies is a two-class support vector machine using multiple kernels. Therefore, it differs from MKAD in the fact that it is not based on a one-class SVM like MKAD, but has the same kernel structure as MKAD. The dual objective function for the two-class problem is:

$$\max_\alpha \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

where $(\mathbf{x}_i, y_i)$'s are the data tuples for $i = 1, \ldots, \ell$. Here $\mathbf{x}_i$ and y_i are the input data points and class labels respectively. In the supervised classification case, the $\mathbf{x}_i$'s correspond to the anomalies found by the MKAD algorithm as discussed above and y_i's correspond to the labels provided by the SMEs. For identifying operationally significant anomalies, this classifier is used to rank the test instances based on their distance from the hyperplane.

2.2 Data Preparation

The surveillance data used in this study comes from combining two Air Traffic Control (ATC) facilities — Denver Terminal Radar Approach Control (D01) and the Denver Air Route Traffic Control Center (ZDV). The objective of this work is to develop a process that automatically discovers previously unmonitored, operationally significant, flight trajectories representing a safety risk to the airspace. The end goal is to produce a tool that can rank these anomalous flights for controllers to review and help make mitigating decisions about the safety of the airspace. The types of anomalies that are being targeted in this study are unusual trajectories from 30 nautical miles (NM) on approach to landing. These can include strange vectoring that do not conform to standard operating procedures, significant overshooting of the final approach fix, or high altitude and speed profiles that can lead to unstable approaches. Fig. 1 illustrates the data processing flow from data collection through merging, filtering, unsupervised anomaly detection, and SME feedback incorporation for classification of anomalies into operationally significant and uninteresting categories. Data collection refers to the process of recording the relevant data that is used in this study (done by the PDARS program responsible for collection, processing, and reporting of aviation data from multiple sources). NASA was given access to PDARS data for the 2014 and 2015 calendar years. Approximately 25,000

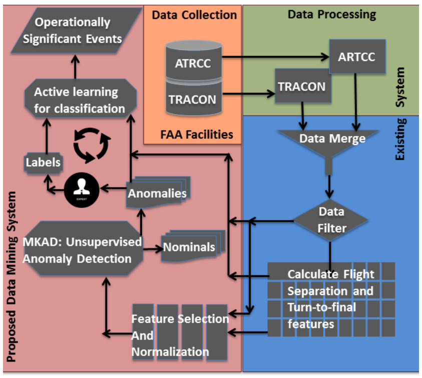

Operationally Significant Events
Data Collection
Data Processing
ATRCC
TRACON
ARTCC
TRACON
Existing System
FAA Facilities
Active learning for classification
Data Merge
Labels
Anomalies
Data Filter
MKAD: Unsupervised Anomaly Detection
Nominals
Calculate Flight Separation and Turn-to-final features
Feature Selection And Normalization
Proposed Data Mining System

Fig. 1. System setup: Data collection, processing, and mining.

flights are available to us from 2014, of which approximately 2400 flights for a particular month are being analyzed as part of our safety study for Denver for 2014. The 2015 flights are only used for validation of results. For each trajectory, from 30 NM out from the destination airport, the minimum separation is found and used to create four-dimensional trajectories: latitude, longitude, altitude and distance to nearest flight. These four features are then averaged over half NM intervals from 30 NM to the runway threshold based on distance traveled and are partitioned by runway and destination airport sets on each day. This results in trajectories with fixed vector lengths because of the half-mile binning and the fixed 30 NM distance traveled, which are then used to create similarity kernels. We also use the PDARS turn-to-final (TTF) reports that provide specific characteristics of how the aircraft performed the turn on to the final approach within 20 NM of a runway. All deviations are calculated with respect to the intercept, which is the point at which the flight trajectory crosses the extended runway centerline before making its final approach. These deviations include intercept distance, angle of intercept, altitude deviation, distance deviation, and speed. Maximum overshoot and aircraft size (categorical feature indicating one of four weight categories) are two additional features from this source. In addition, three binary parameters are derived based on the characteristics of the flight identified as the nearest neighbor for each time step. These features are designed to provide domain context since flights on parallel runways or flights in the same flow are allowed to encroach within the standard separation threshold, whereas flights on the same runway should not fall below the separation threshold. These parameters indicate whether two nearest neighboring flights are on the same runway, parallel runway, or are part of the same flow. An additional derived feature called separation is constructed as the 3-d separation between two flights based on the l_2 norm of the horizontal and vertical separation. It should be noted here that all of these (raw and derived) features together constitute the original feature set for our study. The data is heterogeneous in the sense that some of these features are time-series data while others are a single-point feature and some are continuous whereas others are discrete, nominal, or binary.

The data mining block in Fig. 1 consists of the next steps of unsupervised anomaly detection followed by SME review and labeling, and finally, classifier learning for distinguishing between operationally significant anomalies and uninteresting anomalies. Depending on the size of the input data set, MKAD algorithm may discover hundreds to thousands of ranked anomalies, making it difficult for domain experts to validate all of them. Therefore, we use active learning to learn a classifier using very few labeled instances for this purpose. Each time an SME is provided an instance to be classified, the SME provides the label, along with an explanation/rationale for his/her decision. This rationale, whenever possible, is converted into a new additional feature, which is then incorporated into the classifier training through the creation of a new kernel. The details of this process and approach are described in the next section.

3 Active Learning with Rationales

Active learning algorithms iteratively select informative instances for labeling to save annotation time, cost, and effort [11]. For skewed data sets with minority class distribution much less than the majority class, a common and simple approach for selecting informative instances is to maximize the chances of retrieving positive instances [4]. Most-likely positive (MLP) strategy aims to add more positive instances into the labeled training set. The objective is:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{U}} P_\theta(\hat{\mathbf{y}}^+ | \mathbf{x})$$

where $\hat{\mathbf{y}}^+$ represents the predicted positive label. Intuitively, MLP is a way of over-sampling the minority class to address the imbalanced class distributions. Examples of other strategies include query-by-committee [12], uncertainty sampling [16], expected error reduction [9], evidence-based uncertainty sampling [13], and more. Ramirez-Loaiza et al. [8] provide an empirical evaluation of common active learning strategies. Recent active learning work has looked at eliciting domain knowledge in form of rationales [14] and feature annotations [1] from the SMEs instead of just the labels of instances. In learning with rationales approach, SMEs provide rationales in the form of features that they think are responsible for classifying an instance into a particular class. In this paper, we elicit the rationales from SMEs and incorporate them into the learning process. The main difference between previous work on incorporating rationales and our work is that we create new features based on the rationales provided by the SMEs.

For training our classifier using active learning, we work with the list of anomalies produced by running the unsupervised anomaly detection algorithm, MKAD, on the data described in Section 2.2. For each flight, MKAD returns an anomaly score, which is the flight's distance from the hyperplane of a one-class SVM model. Flights with a negative score are considered as anomalous and flights with a positive score are considered as not anomalous. The SMEs are asked to provide labels for top 5% anomalous flights based on whether they think the anomaly is operationally significant (OS/positive labels) or not (NOS/negative labels). They are also asked to provide a rationale for the chosen label. Since labels and rationales are subjective opinions of each SME, we consolidate the labels and rationales from two SMEs by resolving conflicts (by reviewing each others' labels and rationales) whenever there is one, to get gold standard labels and rationales for our study.

3.1 Creating Rationales

When the SMEs identify a flight as an OS flight, they provide rationales in the form of either domain knowledge or using existing features and thresholds. However, when the SMEs identify a flight as NOS, they only provide acknowledgment of certain characteristics of the flight (e.g., a little overshoot, speed not a factor, small deviations on final). In anomaly detection tasks, it is easy to provide a rationale for why a particular instance is anomalous, but it is often difficult, if not impossible, to provide a rationale for why an instance is not anomalous. Therefore, we use the rationales for only the OS flights to create new features and use them to extend the feature representation. Note that the rationales provided by SMEs are often in terms of the original features that are already captured by PDARS. Some rationales talk about two or more features whereas some highlight only one feature.

In our training set, most OS anomalies could be explained by one or more of three different rationales. The first rationale provided for operational significance is loss-of-separation, which the domain experts define as 'horizontal separation is less than 3 miles and vertical separation is less than a 1000 feet, and the nearest neighboring flight is not on parallel runways and not part of the same flow'. When a loss-of-separation rationale is provided, we create a new feature that checks whether the criteria 'horizontal separation less than 3 miles and vertical separation less than 1000 feet' and 'the nearest neighboring flight is not on parallel runway and not in the same flow' hold and incorporate it as a new binary feature in our training set.

The second rationale provided by the SMEs is for large overshoots where an overshoot is defined as going past a certain point in the landing trajectory against standard operational procedures. For rationales such as 'maximum overshoot is too large', we create a new feature that checks whether the overshoot is greater than a threshold. The threshold can be either chosen manually based on domain knowledge or based on the values of the overshoot feature for the labeled OS flights with overshoot rationale observed until that point, and updated iteratively.

The third rationale provided by the SMEs is for unusual flight path. Since this rationale is more qualitative than quantitative, and none of the original features represent an 'unusual flight path', we compute a new feature as follows. For each runway, using latitude and longitude features, we compute expected flight trajectory as the average trajectory of all flights that land on a runway. Then we create a new feature that captures the overall deviation of each flight from its expected flight trajectory over the last 10 points in the trajectory. Fig. 2 shows the plots for a few trajectories. It can be seen that for the first three flights

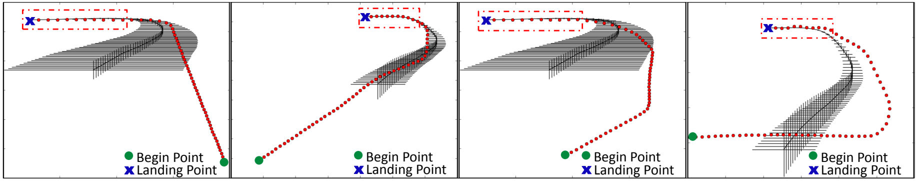


Fig. 2. Expected flight path and deviation from it for 4 flights. The first three flights are NOS. The last flight is an OS flight.

in Fig. 2, the red dots align well with the expected trajectory (highlighted using the red box), whereas for the last flight there is significant deviation from the expected trajectory. This can have severe safety implications and is therefore considered an operationally significant safety event.

3.2 Active Learning with Rationales Algorithm

Algorithm 1 describes our approach for incorporating rationales into active learning. Active learning algorithm starts with a small set of labeled flights, $\mathcal{L}$, and finds the most informative flight, $\mathbf{x}^\star$, from the unlabeled set, $\mathcal{U}$. The most informative flight is the one that provides the classifier maximum information in terms of the decision boundary, or, in other words, one that has the maximum *utility*. The flight $\mathbf{x}^\star$ is then presented to the SME, who provides its label $\mathbf{y}^\star$. For every flight we present to the SME, in addition to a label, we also request for a rationale $R(\mathbf{x}^\star)$ describing why s/he labeled the flight as OS or NOS. If the label is OS, we create a new feature, $f_r^\star$, if possible, for the rationale $R(\mathbf{x}^\star)$ and add it into our existing feature representation: $\mathbf{f} = \langle f_1, f_2, \cdots f_n \rangle \bigcup \langle f_r \rangle$. We assign weight w_o for the original feature kernels and weight w_r for the rationale feature kernels, where $w_r \geq w_o$, since intuitively the rationale features are the ones that have the highest power to separate the OS flights from the NOS ones. However, to satisfy Mercer's condition, we need to ensure that it is a convex combination of the kernels. Therefore, we normalize each weight by the sum of the weights $w = w_o \times n + w_r \times p$, where n and p denote the number of original and rationale features respectively. Let η denote the normalized kernel weights for the enhanced feature set. Note that the kernel weights for original features $\langle \eta_1, \eta_2, \cdots \eta_n \rangle$ are uniform and hence the kernel weight for each original feature will be η_o, which is computed in Step 10 of Algorithm 1. Similarly, the kernel weight for the rationale feature set $\langle \eta_{n+1}, \eta_{n+2}, \cdots \eta_{n+p} \rangle$ is η_r and is computed in Step 11 of Algorithm 1. The final kernel is computed using the updated set of kernel weights η containing normalized weights η_o for the original feature kernels and the normalized weights η_r for the rationale feature kernels for the enhanced feature set $\mathbf{f}$.

Possible enhancements: Based on the training data and the rationales provided by the SMEs, in this paper, we created three features that encompass a significant number of OS safety scenarios. However, this set is far from complete as there can be a huge variety of other explanations that can come from SMEs. So the set of rationale features is always expanding. As the set of features grows based on rationales, there might be a need to consolidate features into conjunctions and disjunctions depending on redundancy. For example, two

Algorithm 1 Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation

```
Input: $\mathcal{U}$ - unlabeled flights, $\mathcal{L}$ - labeled flights, $\mathcal{T}$ - test flights, $\mathbf{f} = \langle f_1, f_2, \cdots f_n \rangle$ -
  current set of features, $\eta = \langle \eta_1, \eta_2, \cdots \eta_n, \eta_{n+1}, \eta_{n+2}, \cdots \eta_{n+p} \rangle$ - normalized kernel
  weights for enhanced feature set, $\theta$ - underlying classification model, $B$ - budget
repeat
  $\mathbf{x}^\star = \arg\max_{\mathbf{x}^i \in \mathcal{U}} utility(\mathbf{x}^i | \theta)$
  request label $\mathbf{y}^\star$ for the flight $\mathbf{x}^\star$
  if $\mathbf{y}^\star ==$ OS then
    request SME to provide a rationale $R(\mathbf{x}^\star)$ for why the flight is operationally
      significant
    if rationale $\neq \phi$ then
      create feature $f_r^\star$ for $R(\mathbf{x}^\star)$
      add $f_r^\star$ to $\mathcal{U}, \mathcal{L}$, and $\mathcal{T}$
      $\eta_o = \frac{w_o}{\sum_{i=1}^{n} \eta_o + \sum_{j=1}^{p} \eta_r}$
      $\eta_r = \frac{w_r}{\sum_{i=1}^{n} \eta_o + \sum_{j=1}^{p} \eta_r}$
      $\eta = \langle \eta_1, \eta_2, \cdots \eta_n \rangle \bigcup \langle \eta_r \rangle$
      $\mathbf{f} = \langle f_1, f_2, \cdots f_n \rangle \bigcup \langle f_r \rangle$
    end if
  end if
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle \mathbf{x}^\star, \mathbf{y}^\star, R(\mathbf{x}^\star) \rangle\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle \mathbf{x}^\star \rangle\}$
  Train $\theta$ on $\mathcal{L}$
until Budget $B$ is exhausted; e.g., $|\mathcal{L}| = B$
```

common rationales in our study are loss-of-separation and large overshoot. However, not all OS flights have both reasons for being labeled OS. Some flights are OS because of loss-of-separation, but they might have perfectly acceptable overshoot values, whereas other OS flights might not have a loss-of-separation but might have large overshoot values. Current framework creates one feature per rationale. An alternative approach is to create one indicator feature and keep revising it by adding the new rationales as disjunctions. Also, once a classifier is trained using this framework, our goal is to find operationally significant events in the original flight data. However, since the classifier is trained on only the anomalies, the feature distribution does not necessarily match that of the overall data set. This unaccounted bias can be handled by sub-sampling some of the flights that are not signaled by MKAD and adding them to the training with NOS (negative) labels. Selecting flights that are ranked lowest by MKAD, for this purpose, can ensure with a high probability that the flights which are most certainly nominal are being used as NOS samples.

4 Empirical Evaluation

Experimental Setup: The data set used for training the classifier using active learning corresponds to PDARS data from the Denver Airport for August

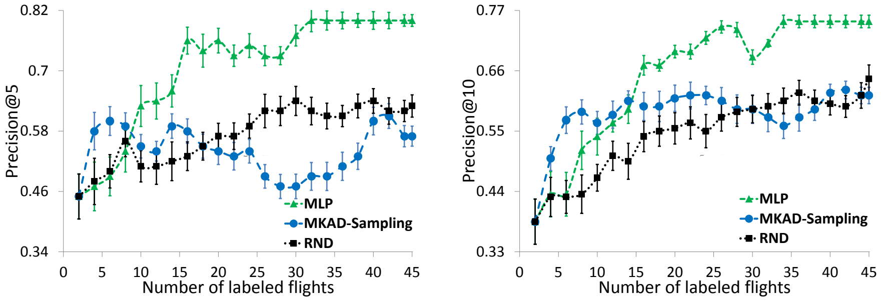
2014, containing approximately 2400 flights out of which 153 flights are marked anomalous by MKAD. These 153 flights are reviewed by two SMEs independently (with conflict resolutions as needed) to provide labels and explanations. In these 153 flights, 26 are marked OS (positive) and the remaining 127 are marked NOS. The original data set contains 16 features as described in Section 2.2. Additionally, we construct 3 rationale features supporting the explanations for the OS flights during the active learning iterations, when OS flights with one or more rationales provided in Section 3.1 are encountered.

Our proposed active learning strategy, MLP_w/Rationales, selects most-likely positive (MLP) instances for labeling at each iteration of training and creates (or updates) rationale features whenever an appropriate new instance is encountered. We compare our algorithm's performance with three baselines: (i) random strategy (RND) where random instances are picked from the unlabeled pool and given to the SME for labeling, (ii) most-likely positive strategy (MLP) that selects more of the positive instances for labeling at each iteration, but does not add new features (or rationales), and (iii) MKAD-Sampling strategy where flights are given to the SME for labeling in the order of their MKAD anomaly ranking (higher the anomaly rank, the more informative it is for labeling).

We evaluate all strategies using $precision@k$ measure which can be defined as the number of positive instances in top k instances ranked by the classifier. This measure is most suitable for our application because the SMEs go through a list of anomalies to identify those that are operationally significant for further investigation, and improving $precision@k$ means that the SMEs would analyze more of the OS flights compared to the NOS flights. We chose $precision@5$ and $precision@10$ for evaluation since they are the most frequently used in the literature measures to use (e.g., [3],[17]). We bootstrap the classifier using an initially labeled set containing one OS flight and one NOS flight, and at each round of active learning the learner picks a new flight for labeling. We evaluate all strategies using 2-fold cross validation and repeat each experiment 10 times per fold starting with a different bootstrap, and present average results over 20 different runs. We set the budget (B) in our experiments to 45 flights, as most learning curves flatten out after about 35 flights. Since each learning curve is an average over 20 runs, for each learning curve, we report error bars for standard error of the mean (SEM), which is computed as standard deviation divided by the square root of sample size ($SEM = \frac{s}{\sqrt{n}}$).

4.1 Results

Fig. 3 presents the learning curves comparing RND, MKAD-Sampling, and MLP strategies for $precision@5$ and $precision@10$. MKAD-Sampling performs worse than RND for $precision@5$ and it outperforms RND for $precision@10$. However, MLP outperforms both RND and MKAD-Sampling for $precision@5$ and $precision@10$. We performed pairwise one-tailed t-tests under significance level of 0.05, where pairs are area under the learning curves for 20 runs of each method. If a method has higher average performance than a baseline with a significance level of 0.05 or better, it is a win, if it has significantly lower performance,



(a) (b)

Fig. 3. MLP vs. RND and MKAD-Sampling. MLP significantly outperforms RND and MKAD-Sampling for both (a) *precision*@5 and (b) *precision*@10.

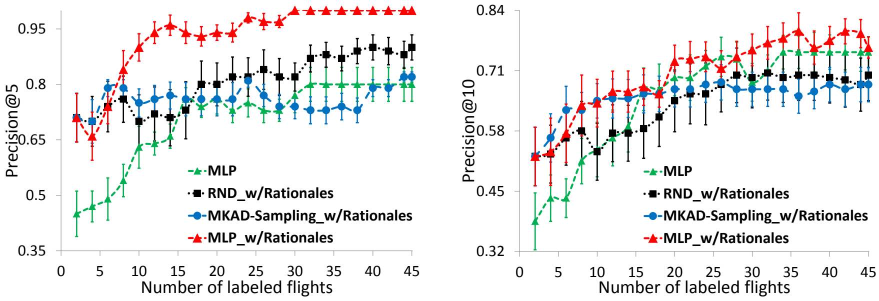
it is a loss, and if the difference is not statistically significant, the result is a tie. The t-test results show that MKAD-Sampling statistically significantly loses to RND for *precision*@5 and significantly wins over RND for *precision*@10. MKAD-Sampling performs better than MLP at the very beginning of the learning curves, but t-test results show that overall, MLP statistically significantly wins over MKAD-Sampling for both *precision*@5 and *precision*@10. This justifies our choice of using MLP as the active learning strategy for training our classifier for a highly skewed distribution of class labels.

Table 1 presents a comparison of the number of labeled flights required by these methods to achieve a target value of *precision*@5 and *precision*@10. The maximum target for each metric is chosen based on the best performance observed in the learning curves for each of the strategies. The results show that MLP often requires fewer labeled flights compared to RND and MKAD-Sampling. Moreover, MLP achieves a *precision*@5 of 0.7 and *precision*@10 of 0.65 with just 16 labeled flights, whereas RND and MKAD-Sampling could not achieve these targets even with 45 labeled flights.

Next, we present the results that demonstrate the effect of incorporating rationales into active learning. Fig. 4 presents the learning curves comparing MLP strategy for active learning without rationales (MLP) and MLP with rationales strategy (MLP_w/Rationales) that utilizes MLP to select instances and incorporates rationales iteratively during active learning (refer to Algorithm 1).

Table 1. Comparison of number of labeled flights required by various strategies to achieve a target performance measure. 'n/a' represents that the target performance cannot be achieved by a method even with 45 labeled flights.

	Target *precision*@5						Target *precision*@10					
Method	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**0.50**	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**
RND	6	25	n/a	n/a	n/a	n/a	12	18	33	n/a	n/a	n/a
MKAD-Sampling	4	6	n/a	n/a	n/a	n/a	4	6	13	n/a	n/a	n/a
MLP	5	10	16	32	n/a	n/a	8	12	15	16	23	34
MLP_w/Rationales	2	2	2	8	10	29	2	5	7	11	19	29



(a) (b)

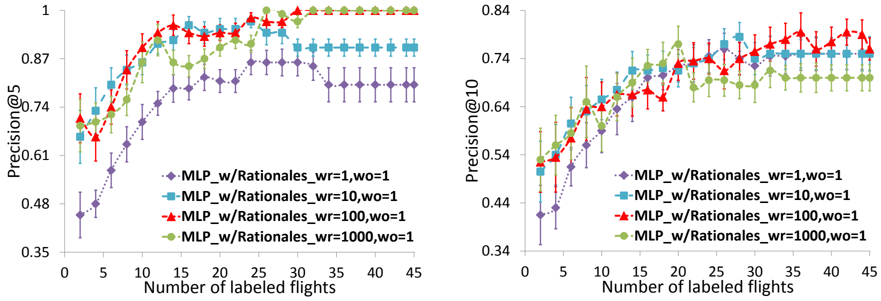
Fig. 4. MLP_w/Rationales vs. MLP. Incorporating rationales further improves performance over MLP for both (a) *precision*@5 and (b) *precision*@10.

We set the rationale feature weight $w_r = 100$ and the original feature weight, $w_o = 1$. The results show that MLP_w/Rationales statistically significantly wins over MLP for both *precision*@5 and *precision*@10 performance measures. Moreover, MLP_w/Rationales requires even fewer labeled flights compared to MLP to achieve the same target performance measure, as shown in Table 1. For example, MLP achieves a target *precision*@5 of 0.8 with 32 labeled flights, whereas MLP_w/Rationales achieves this target with only 8 labeled flights, which is 75% savings in the labeling effort over MLP.

Fig. 4 also compares MLP_w/Rationales to RND_w/Rationales and MKAD-Sampling_w/Rationales. MKAD-Sampling_w/Rationales performs better than MLP_w/Rationales at the beginning for both *precision*@5 and *precision*@10, but after seeing approximately 10 labeled instances, MLP_w/Rationales outperforms MKAD-Sampling_w/Rationales. T-tests show that MLP_w/Rationales statistically significantly outperforms both MKAD-Sampling_w/Rationales and RND_w/Rationales for both *precision*@5 and *precision*@10.

Choice of rationale weights: We ran experiments to study the effect of weights w_r and w_o on the performance of our algorithm. We chose uniform weighting for the original feature kernels since all 16 of those were suggested by domain experts and were supposed to be important for this safety study. We fixed w_o=1 and experimented with four weight settings for w_r (1, 10, 100, or 1000). Fig. 5 presents the learning curves for these four weight settings for MLP_w/Rationales. The results confirm our intuition that weighting rationale features higher than original features provides benefit to the active learner. The *precision*@5 results are significantly better with w_r=100 than other weights for w_r. For *precision*@10, setting higher weights for rationale features improves performance at the beginning of active learning, however, t-test results show that weights w_r=1, 10, and 100 statistically significantly tie with each other. In general, weighting rationale features higher than original features improves learning. The kernel weights for optimal performance can be obtained through multiple kernel learning.

Ideally, one would want to search for the best weights setting using cross validation, but given the limited number of anomalous instances that domain experts could review, it was not possible for us to perform cross validation over



(a) (b)

Fig. 5. Comparison of rationale features weights w_r for MLP_w/Rationales using (a) *precision*@5 and (b) *precision*@10

the training set. Based on the performance observed for these four weight settings, we chose w_o=1 and w_r=100 for all our experiments.

Scalability: Active learning methods are typically computationally expensive, since they need to build a classifier at each iteration of learning and evaluate the utility score for every instance in the unlabeled pool. However, in our setting, when active learning is used on the output of an unsupervised anomaly detection algorithm, the unlabeled pool is much smaller in size compared to the entire set of raw instances. Therefore, utilizing this framework in a practical setting is easily viable, without the iterative nature of active learning being a performance bottleneck.

4.2 Performance Benefits

In the absence of active learning framework, our SMEs took approximately 33 hours to review the entire set of 153 anomalies produced by MKAD. These 33 hours were spread over multiple weeks due to limited availability of SME time for such tasks, which is a standard problem in the industry. As Fig. 4 shows, most of the learning curves flatten out after labeling 35 flights. This would reduce the SME review time to less than one-third of the original time. This has implications on both man-hours and monetary savings. Moreover, active learning with state-of-the-art (MKAD-Sampling) achieves *precision*@5 of 0.57 and *precision*@10 of 0.61. Active learning with rationales (MLP_w/Rationales) achieves *precision*@5 of 1 (75.4% improvement over MKAD-Sampling) and *precision*@10 of 0.76 (24.6% improvement over MKAD-Sampling).

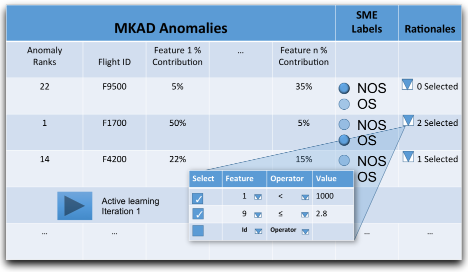
Validation set results: Currently, MKAD is being used as an unsupervised anomaly detection method to find statistically significant anomalies in the data. We compare performance benefits that active learning with rationales framework (MLP_w/Rationales) provides over the MKAD based classifier for finding OS anomalies in two external validation data sets, July 2014 and July 2015 data sets for the Denver airport. The July 2014 data set has 149 labeled flights with 24 OS anomalies and July 2015 data set has 257 labeled flights with 84 OS anomalies, as determined by the SMEs. Both *precision*@5 and *precision*@10 values for MKAD are 0.4 for the July 2014 data set, and 0.2 for the July 2015 data set. Using our (MLP_w/Rationales) framework, *precision*@5 improves by 15% for July 2014 data set and by 50% for July 2015 data set. On the other hand, *precision*@10 improves by 25% and 110% for the July 2014 and July 2015 data sets, respectively.

It should be noted that MKAD performs very poorly for the July 2015 data set. This is because the data set is expected to evolve significantly over the years (due to change in landing procedures and other regulation changes) and the MKAD classifier does not capture the signatures of the OS flights, but rather focuses on finding statistically different data points which can vary over time due to a change in the underlying distribution. However, the nature of the operationally significant anomalies still remains consistent and therefore MLP_w/Rationales can identify those types of anomalies much better than MKAD. These results show how active learning with rationales framework can help in building a classifier that is robust to changing distribution of statistically significant anomalies and can, therefore, be used on new data sets without further labeling needs.

5 Towards Deployment

The active learning framework improves over traditional learning, and incorporating rationales further improves learning, utilizing the SME's time much more efficiently. The classifier that is trained through this framework is focused on finding operationally significant anomalies, rather than simply statistically significant anomalies, and hence the flights that are signaled by the two-class classifier approach are of higher relevance to FAA.

This active learning framework has been developed as an extension to the anomaly detection framework that is currently used for detecting safety events. We expect this framework to easily fit into the existing anomaly detection framework because the classifier training is part of the same data flow pipeline that can take the output of MKAD as input and can seamlessly plug-in new data sources as needed. Given that the new classifier reduces SME review time significantly while improving coverage and reducing false alarm rate, it seems to be the perfect addition to bolster the existing anomaly detection framework, especially since these safety studies are conducted on a regular basis on data that gets collected every month. We expect that this enhanced data processing pipeline with the active learning framework incorporated into it will make the review and detection system significantly more efficient. In our current setup, we provide our SMEs an excel sheet containing the list of anomalies returned by MKAD and the SMEs note down the annotations and rationales textually. This process is repeated iteratively for each round of labeling. The textual information is then converted into features in batches. The next step towards the deployment of our active learning with rationales framework is to fully automate this process where the SMEs can select appropriate rationales using a drop-down list of features by choosing the criteria that were satisfied or violated by the flight in question. The SMEs can choose multiple features for each flight and, therefore, create complex rationale



MKAD Anomalies					SME Labels	Rationales
Anomaly Ranks	Flight ID	Feature 1 % Contribution	...	Feature n % Contribution		
22	F9500	5%		35%	● NOS ○ OS	0 Selected
1	F1700	50%		5%	○ NOS ● OS	2 Selected
14	F4200	22%		15%	○ NOS OS	1 Selected
Active learning Iteration 1						
...	...				...	...

Select	Feature	Operator	Value
☑	1	<	1000
☑	9	≤	2.8
□	Id	Operator	

Fig. 6. Diagrammatic representation of the GUI for deployment of active learning as part of the anomaly detection framework

conditions that can be used to create new complex discriminative features on the fly and those features can be immediately utilized for the next iteration of active learning. Fig. 6 shows a diagrammatic representation of the software that we are currently developing for deploying as part of the existing framework. It shows the SME initial bootstrap instances for labeling by randomly selecting from the list of anomalies found by MKAD, along with the feature contributions and asks for labels and rationales using drop-down menus. As soon as the classifier has enough number of bootstrap samples, training begins for the classifier. After every iteration the most informative instance is populated in the table for the SME to label and rationalize and classifier training begins again. This iterative process is repeated until the budget B is exhausted or there is no further improvement in the classifier performance on a held-out set.

6 Conclusion

We present an active learning framework to build a classifier that can distinguish between operationally significant anomalies and uninteresting ones. Our proposed framework is novel in the sense that it incorporates SME feedback into the learning process in the form of new features constructed to support the labels. Experimental evaluation on real aviation data shows that our approach improves detection of operationally significant events by as much as 75% compared to the state-of-the-art. The learnt classifier also generalizes well when tested on additional validation data sets. We also observe that our approach provides significant reduction in SME review time and labeling effort in order to achieve the same target performance using other baselines.

We are working toward deploying our framework as a daily reporting system that can reveal operationally significant anomalies to safety analysts with the goal of developing mitigation opportunities by changing standard operating procedures. The reduced false alarm rate of our framework compared to the unsupervised anomaly detection method is critical for domain experts to accept our reporting system and not just ignore the alarms, as has happened with other warning systems. Future work also includes developing richer rationales and ability to integrate multiple data sources for supporting those rationales for increased coverage of a wider range of operationally significant anomalies.

Acknowledgments. This research is supported by the NASA Airspace Operation and Safety Program. Manali Sharma and Mustafa Bilgic are supported by the National Science Foundation CAREER award no. IIS-1350337. The authors would also like to thank the SMEs: Steve Wyloge and Glenn Hilgedick for their insightful comments and perspective on the identified events.

References

1. Attenberg, J., Melville, P., Provost, F.: A unified approach to active dual supervision for labeling features and examples. In: ECML. pp. 40–55 (2010)
2. Bach, F., Lanckriet, G., Jordan, M.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML (2004)
3. Bharat, K., Henzinger, M.R.: Improved algorithms for topic distillation in a hyperlinked environment. In: ACM SIGIR. pp. 104–111. ACM (1998)
4. Bilgic, M., Bennett, P.N.: Active query selection for learning rankers. In: ACM SIGIR (August 2012)
5. Das, S., Matthews, B.L., Srivastava, A.N., Oza, N.C.: Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In: Proceedings of KDD. pp. 47–56 (2010)
6. National Research Council: Advancing Aeronautical Safety: A Review of NASA's Aviation Safety-Related Research Programs. The National Academies Press, Washington DC (2010)
7. Pelleg, D., Moore, A.: Active learning for anomaly and rare-category detection. In: NIPS (December 2004)
8. Ramirez-Loaiza, M.E., Sharma, M., Kumar, G., Bilgic, M.: Active learning: an empirical study of common baselines. Data Mining and Knowledge Discovery pp. 1–27 (2016)
9. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: ICML. pp. 441–448 (2001)
10. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation 13(7), 1443–1471 (2001)
11. Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2012)
12. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: ACM Annual Workshop on Computational Learning Theory. pp. 287–294 (1992)
13. Sharma, M., Bilgic, M.: Evidence-based uncertainty sampling for active learning. Data Mining and Knowledge Discovery pp. 1–39 (2016)
14. Sharma, M., Zhuang, D., Bilgic, M.: Active learning with rationales for text classification. In: NAACL-HLT. pp. 441–451 (2015)
15. Sindhwani, V., Melville, P., Lawrence, R.D.: Uncertainty sampling and transductive experimental design for active dual supervision. In: ICML. pp. 953–960 (2009)
16. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. JMLR 2, 45–66 (2001)
17. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: Proceedings of the 26th annual international conference on machine learning. pp. 1169–1176. ACM (2009)
18. Zaidan, O.F., Eisner, J., Piatko, C.: Machine learning with annotator rationales to reduce annotation cost. In: Proceedings of the NIPS* 2008 Workshop on Cost Sensitive Learning (2008)