Incorporating Expert Feedback into Active Anomaly Discovery

Shubhomoy Das, Weng-Keen Wong, Thomas Dietterich, Alan Fern, and Andrew Emmott
Oregon State University
Corvallis, OR, USA
{dassh, wongwe, tgd, alan.fern, emmotta}@oregonstate.edu

***Abstract*—Unsupervised anomaly detection algorithms search for outliers and then predict that these outliers are the anomalies. When deployed, however, these algorithms are often criticized for high false positive and high false negative rates. One cause of poor performance is that not all outliers are anomalies and not all anomalies are outliers. In this paper, we describe an Active Anomaly Discovery (AAD) method for incorporating expert feedback to adjust the anomaly detector so that the outliers it discovers are more in tune with the expert user's semantic understanding of the anomalies. The AAD approach is designed to operate in an interactive data exploration loop. In each iteration of this loop, our algorithm first selects a data instance to present to the expert as a potential anomaly and then the expert labels the instance as an anomaly or as a nominal data point. Our algorithm updates its internal model with the instance label and the loop continues until a budget of B queries is spent. The goal of our approach is to maximize the total number of true anomalies in the B instances presented to the expert. We show that when compared to other state-of-the-art algorithms, AAD is consistently one of the best performers.**

I. Introduction

High accuracy anomaly detection algorithms have the potential to solve difficult problems in many application domains, including insider threat detection, biosurveillance, computer security, data cleaning and scientific discovery. The goal of anomaly detection is to identify unusual data instances of interest. More precisely, we define an *anomaly* to be a data instance that is generated by a process that is different from the process generating the "normal" data instances, which we refer to as *nominal* instances. Anomalies typically constitute a small percentage of the data (e.g. 1% or even less).

Since known anomalies are scarce, most anomaly detection algorithms (eg. [1], [2], [3], [4]) are applied in an unsupervised setting in which an unlabeled data set is used to build a model of nominal data instances, even though the data may be contaminated by a small percentage of anomalies. The top B outlier instances under the nominal data model are then identified as anomaly candidates. B is determined by some combination of the expected anomaly rate and the amount of human resources available to investigate the anomaly candidates. This approach, however, usually leads to high false positive and false negative rates.

One cause of poor performance is that not all outliers are anomalies and not all anomalies are outliers. If the distribution of nominal data points has heavy tails, then many outliers will be nominal. Conversely, in adversarial situations, the adversary is trying to mimic the nominal data points, so the anomalies will be buried in regions of high nominal density. There is no statistical solution to these problems—the designer of the anomaly detection system must choose features to mitigate these problems. Another cause of poor performance is the difficulty of performing any kind of feature selection in unsupervised anomaly detection. As most anomaly detection applications involve high-dimensional feature spaces, this leads inexorably to poor anomaly detection performance. In this paper, we describe a method for incorporating expert feedback to adjust the anomaly detector so that it puts more weight on relevant features and ignores features that do not correspond to the expert's semantic understanding of the anomalies.

We consider an interactive data exploration loop. Initially, the anomaly detector is applied to an unlabeled dataset. Then, the anomaly detector presents a data instance to the analyst and asks the expert to label it as anomalous or nominal. The analyst labels this instance and the anomaly detector updates its model with the newly acquired instance labels. The process resumes with the next iteration presenting another data instance to the analyst. This process continues until a budget B on the total number of instances presented to the analyst is spent. Our goal is to maximize the number of true anomalies presented to the analyst. For most applications, we expect $B < 100$.

II. Related Work

We call our approach *Active Anomaly Discovery* (AAD) to differentiate it from active learning [5]. In active learning, the goal is to select the most informative instances for an analyst to label with the aim of training a classifier that has the highest predictive performance on unseen data instances. In contrast, the performance of AAD is not measured by the predictive ability of the classifier on an unseen test set, but rather by the total number of anomalies presented to the analyst after the interactive data exploration loop finishes. Active learning has been applied to anomaly detection [6], [7], [8], [9] using a variety of query strategies. A common query strategy (also used in our approach) is to ask the user to label the instances which are least likely under the current model [6], [9]. Another strategy combines uncertainty sampling with querying the most likely anomalies [7], [8].

Rare category detection (RCD) [10], a field related to anomaly detection, tries to identify a representative of each class with as few queries as possible. The user either identifies an instance as belong to one of K existing classes or to a new, previously unseen class. AAD can be viewed as a simpler version of RCD; it places a lighter cognitive burden on the analyst by only requiring instances to be labeled as *anomaly* or *nominal* rather than placing them into K distinct classes.

Our work is also closely related to Learning to Rank (LTR) [11]. In LTR, the goal is to learn a ranking function that can return a total (or partial) order over a set of instances. One strategy for LTR is based on learning from pairwise preferences [12], [13], which is a technique that we adapt to our approach. Recently, loss functions for maximizing accuracy at the top of the ranked list [14], [15], [16] have been proposed and we use these loss functions in our work.

III. Methodology

In our setting, we are given a dataset $\mathbf{D}$ with n instances, $\mathbf{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$. Each instance i is a tuple $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is a d-dimensional real vector and $y_i \in \{\textit{anomaly}, \textit{nominal}\}$ is the (hidden) class label. We assume that there is a large class imbalance, with $y_i = \textit{nominal}$ comprising the overwhelming majority of the data.

A subset of the class labels is progressively revealed as the analyst provides the class labels during the interactive feedback loop. In the first iteration, the model chooses a data instance $\mathbf{x}_1^q$ to present to the analyst, and the analyst provides a class label y_1^q for the queried instance. This process continues until a total of B instances are queried. We represent the sequence of queries as $\mathbf{X}_B^q = (\mathbf{x}_1^q, \mathbf{x}_2^q, \ldots, \mathbf{x}_B^q)$. The class labels associated with these instances are provided as feedback by the analyst. We denote the analyst feedback as $\mathbf{F} = ((\mathbf{x}_1^q, y_1^q), (\mathbf{x}_2^q, y_2^q), \ldots, (\mathbf{x}_B^q, y_B^q))$.

The goal of our work is to maximize the number of true anomalies seen by the analyst over the B total instances presented:

$$\arg\max_{\mathbf{X}_B^q} |\{(\mathbf{x}_i^q, y_i^q) \in \mathbf{F} : y_i^q = \textit{anomaly}\}| \tag{1}$$

Equation 1 cannot be computed because we do not know the true class labels. To achieve our goal, we greedily select instances to query that our model assigns the highest probability of being an anomaly. Even though our approach is greedy, we show that it is very effective in our empirical results.

A. Anomaly Detector

For our anomaly detector, we employ the *Loda* algorithm [4], which is an ensemble $\mathbf{P} = \{\mathbf{p}_m\}_{m=1}^M$ of M one-dimensional histogram density estimators computed from sparse random projections. Each projection $\mathbf{p}_m$ is defined by a sparse d-dimensional random vector $\boldsymbol{\beta}_m$, with $1/\sqrt{d}$ randomly chosen non-zero components, with each value of these non-zero components drawn from a standard normal distribution. Loda constructs a one-dimensional density estimator f_m by projecting each data point onto the real line according to $\mathbf{p}_m(\mathbf{x}_i) = \boldsymbol{\beta}_m^\top \mathbf{x}_i$ and forming a histogram density estimator f_m. The anomaly score assigned to point $\mathbf{x}_i$ is the mean negative log density (mean surprise):

$$a_i^{Loda} = \frac{1}{M} \sum_{m=1}^{M} -\log(f_m(\mathbf{x}_i)) \tag{2}$$

We can view Loda as a representation transformation that converts a data point $\mathbf{x}_i$ in the d-dimensional feature space into a log probability vector in M-dimensional real space. Denote the latter as $\mathbf{z}_i = [-\log(f_1(\mathbf{x}_i)), ..., -\log(f_M(\mathbf{x}_i))]^T$, where $\mathbf{z}_i \in \mathbb{R}^M$. The negative-log-pdf for the entire dataset will be represented by $\mathbf{H} = [\mathbf{z}_1, ..., \mathbf{z}_n]^\top$. With this notation and defining $\mathbf{w}_U = [\frac{1}{M}, ..., \frac{1}{M}]^T \in \mathbb{R}^M$, we can write Equation 2 in a more compact form: $a_i^{Loda} = \mathbf{w}_U \cdot \mathbf{z}_i$.

Loda gives equal weights to all projections and since these projections are selected at random, it is not guaranteed that every projection is good at isolating anomalies by itself. Once the projections have been computed by Loda and fixed, we propose to integrate analyst feedback by learning a better weight vector $\mathbf{w}$ that assigns higher weights to the more useful projections and lower to the less useful ones. Our approach is not restricted to using LODA. Other ensemble methods based on random projections (e.g. [17]) could also be employed.

B. The top τ-quantile

Internally, our model maintains a list of data instances ranked by the anomaly score produced by the anomaly detector. We wish to keep the labeled anomalies at the top of the ranked list, where the top of the list is defined as the top τ-quantile. The top τ-quantile value of a function $h : \mathcal{X} \to \mathbb{R}$ is defined as the value $q_\tau : P_x(h(x) > q_\tau) = \tau$.

For any $\tau \in [0, 1]$, let ρ_τ be defined as:

$$\forall u \in \mathbb{R},\ \rho_\tau(u) = -\tau(u)_- + (1-\tau)(u)_+$$
$$\text{where, } (u)_+ = \max(u, 0), \text{ and } (u)_- = \min(u, 0)$$

The τ quantile value q_τ of a sample of real numbers $\{u_1, ..., u_n\}$ can be computed as $q_\tau = \arg\min_{u \in \mathbb{R}} \sum_{i=1}^n \rho_\tau(u_i - u)$ [18]. This method can be useful when the quantile value needs to be computed simultaneously as part of an optimization function.

An alternative (and natural) way to compute the top τ-quantile value is to sort all values in descending order and select the τn-th ranked value.

C. Accuracy at the top

We want the scores of all labeled anomalies to be higher than q_τ and the scores of all labeled nominals to be below q_τ. When this property is violated on a specific data instance $(\mathbf{z}_i, y_i)$, we incur the loss shown in Equation 3, where $\mathbf{w}$ is a vector of weights. Equation 3 is based on the surrogate

empirical loss function defined in the Accuracy at the Top (AATP) approach [14].

$$\ell(q_\tau, \mathbf{w}; \mathbf{z}_i, y_i) = \begin{cases} 0 & \mathbf{w}\cdot\mathbf{z}_i \geq q_\tau \text{ and } y_i = `anomaly' \\ 0 & \mathbf{w}\cdot\mathbf{z}_i < q_\tau \text{ and } y_i = `nominal' \\ (q_\tau - \mathbf{w}\cdot\mathbf{z}_i) & \mathbf{w}\cdot\mathbf{z}_i < q_\tau \text{ and } y_i = `anomaly' \\ (\mathbf{w}\cdot\mathbf{z}_i - q_\tau) & \mathbf{w}\cdot\mathbf{z}_i \geq q_\tau \text{ and } y_i = `nominal' \end{cases} \tag{3}$$

The *AATP* approach in [14] was presented in a supervised learning setting, meaning that it typically expects all pair-wise preferences between relevant and irrelevant items to be available during training. In contrast, all pairwise preferences between anomalies and nominals are not available in our initial dataset and we typically only obtain a small subset of these pairwise preferences when instance labels are revealed. As a result, the loss in Equation 3 needs to be computed by summing up the loss over the instances in the labeled feedback set $\mathbf{H}_F$, i.e., the instances labeled by the analyst. The weights $\mathbf{w}$ that minimize the overall loss can be computed as:

$$\begin{aligned} \mathbf{w} &= \arg\min_{\mathbf{w}} \left(\sum_{\mathbf{z}_i \in \mathbf{H}_F} \ell(q_\tau, \mathbf{w}; (\mathbf{z}_i, y_i)) \right) \\ \text{s.t., } q_\tau &= \arg\min_{u} \sum_{\mathbf{z}_i \in \mathbf{H}} \rho_\tau(\mathbf{w}\cdot\mathbf{z}_i - u) \end{aligned} \tag{4}$$

Equation 4 is not convex because the equality constraint is not affine [14]. This makes joint inference of $\mathbf{w}$ and q_τ hard. As an alternative, we solve for $\mathbf{w}$ and q_τ separately. The steps are shown below, with $\mathbf{w}^{(t-1)}$ denoting the value of $\mathbf{w}$ from iteration $(t-1)$.

1) Solve $q_\tau = \hat{q}_\tau(\mathbf{w}^{(t-1)})$ using the (fixed) value of $\mathbf{w}^{(t-1)}$. We compute $\hat{q}_\tau(\mathbf{w}^{(t-1)})$ by ranking the anomaly scores given $\mathbf{w}^{(t-1)}$.
2) Compute $\mathbf{w}^{(t)}$ using Equation 4 keeping $q_\tau = \hat{q}_\tau(\mathbf{w}^{(t-1)})$ fixed.

Under this approach, there is no guarantee that $\hat{q}_\tau(\mathbf{w}^{(t-1)})$ still remains the τ-th quantile score under the new value $\mathbf{w}^{(t)}$. Nevertheless, this approximation was performed in [14] and shown to produce reasonable results.

D. Objective Function

A straightforward application of the AATP approach produces the objective function in Equation 4. However, even with the approximation with $\hat{q}_\tau(\mathbf{w})$, Equation 4 needs to be further modified in order to satisfy our goal. Our overall objective function is shown in Equation 5 and we discuss the four necessary modifications below.

The modifications to Equation 4 are as follows. First, we divide the labeled dataset $\mathbf{H}_F$ into the set of labeled anomalies $\mathbf{H}_A$ and the set of labeled nominals $\mathbf{H}_N$. We then introduce a weight C_A that causes the loss for anomalies in $\mathbf{H}_A$ to be higher than that associated with nominals.

Second, we use soft pair-wise constraints [12] to encourage labeled anomalies to have higher scores than labeled nominals. Since we assume that the number of instances labeled by a user would be limited by a small budget (e.g., < 100 queries), the resulting optimization can be solved in under a few minutes of running time on most modern computers.

Third, the proximal factor $\|\mathbf{w} - \mathbf{w}_p\|^2$ avoids a degenerate solution of $\mathbf{w} = 0$ and keeps the learned weights close to the uniform weights of Loda. The Loda algorithm has been shown to perform well as an anomaly detector [4] and we want to adjust but not make drastic changes to an otherwise good model.

Finally, we introduce the τ-th ranked instance (under the current model) as a *proxy* anomaly when there are only labeled nominals and no labeled anomalies. This ensures that the known nominals and instances which are similar to them are forced lower in ranking due to pair-wise constraints while other potential anomalous instances rise to the top.

$$\begin{aligned} \mathbf{w}^{(t)} = \arg\min_{\mathbf{w},\xi} & \frac{C_A}{|\mathbf{H}_A|} \left(\sum_{\mathbf{z}_i \in \mathbf{H}_A} \ell(\hat{q}_\tau(\mathbf{w}^{(t-1)}), \mathbf{w}; (\mathbf{z}_i, y_i)) \right) \\ & + \frac{1}{|\mathbf{H}_N|} \left(\sum_{\mathbf{z}_i \in \mathbf{H}_N} \ell(\hat{q}_\tau(\mathbf{w}^{(t-1)}), \mathbf{w}; (\mathbf{z}_i, y_i)) \right) \\ & + \|\mathbf{w} - \mathbf{w}_p\|^2 + C_\xi \sum_{i,j} \xi_{ij} \end{aligned} \tag{5}$$

s.t.,

$$(\mathbf{z}_i - \mathbf{z}_j)\cdot\mathbf{w} + \xi_{ij} \geq 0 \quad \forall \mathbf{z}_i, \mathbf{z}_j : \mathbf{z}_i \in \mathbf{H}'_A, \mathbf{z}_j \in \mathbf{H}_N,$$
$$\xi_{ij} \geq 0$$

where, $\mathbf{w}_p = \frac{\mathbf{w}_U}{\|\mathbf{w}_U\|} = [\frac{1}{\sqrt{m}}, \ldots, \frac{1}{\sqrt{m}}]^T$, $\hat{q}_\tau(\mathbf{w}^{(t-1)})$ is computed by ranking anomaly scores using $\mathbf{w}^{(t-1)}$ and,

$$\mathbf{H}'_A = \begin{cases} \{\mathbf{x}_\tau\} & \text{when } \mathbf{H}_A = \emptyset \\ \mathbf{H}_A & \text{when } \mathbf{H}_A \neq \emptyset \end{cases}$$

($\mathbf{x}_\tau$ is the τ-th quantile *proxy* anomaly instance)

E. Active Anomaly Discovery (AAD)

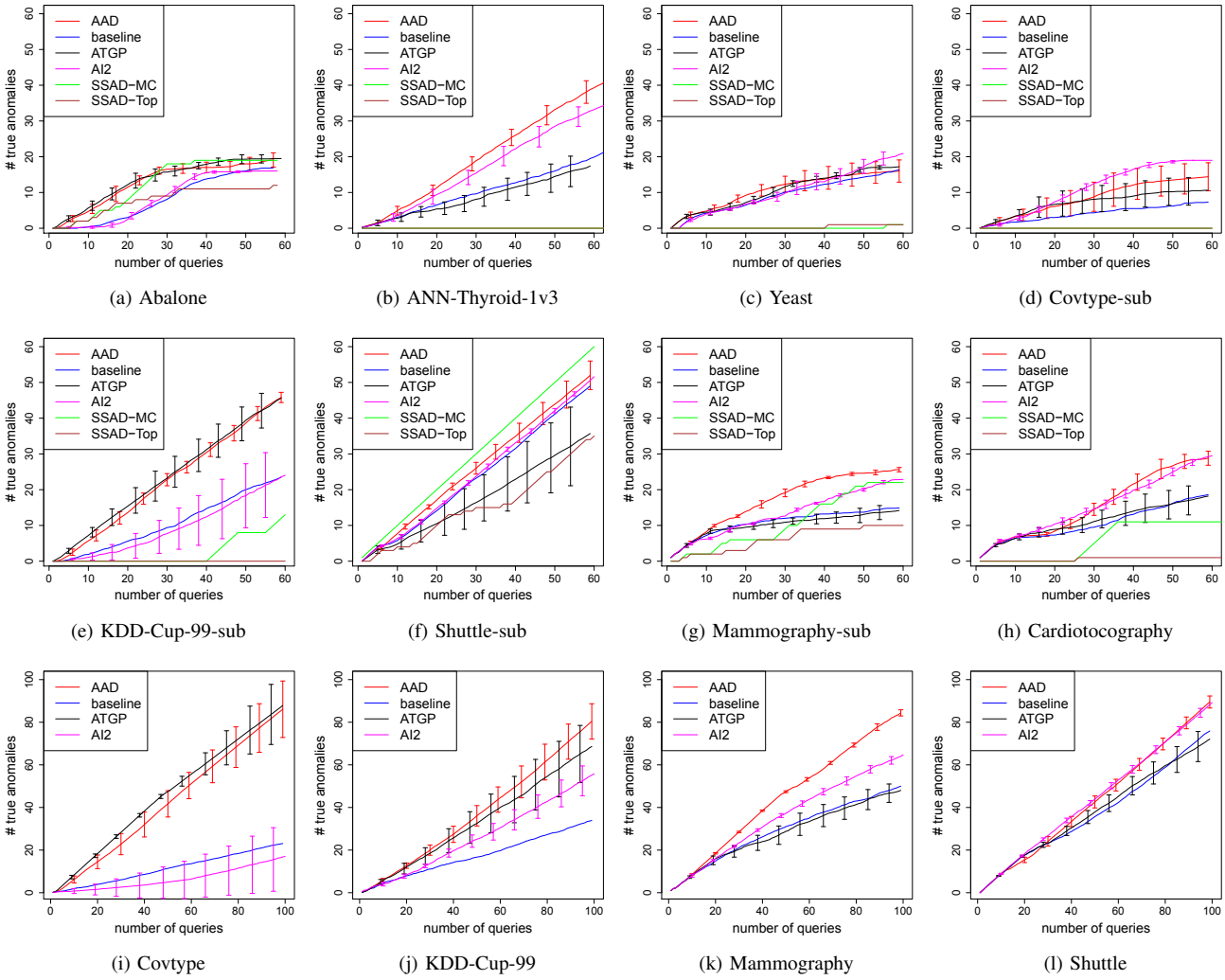
Algorithm 1 describes the active learning loop for our approach. Note that we ask the analyst to label the top ranked instance that has not already been labeled.

IV. Results

A. Experimental Setup

Since our objective is to maximize the number of true anomalies presented to the analyst, we plot the total number of true anomalies discovered against the number of queries presented to the user. We call this plot an *anomaly discovery curve*. Ideally, this curve climbs as quickly as possible. In our experiments we assume that the user is an *oracle* that labels instances correctly. We compare the results of our proposed algorithm against four other algorithms on nine different datasets. The algorithms we compare against are:

1) Baseline: For the baseline, we present instances in decreasing order of anomaly score computed with the original Loda algorithm (i.e. with uniform weights). This baseline captures the performance of an unsupervised anomaly detector that does not incorporate expert feedback.



(a) Abalone (b) ANN-Thyroid-1v3 (c) Yeast (d) Covtype-sub

(e) KDD-Cup-99-sub (f) Shuttle-sub (g) Mammography-sub (h) Cardiotocography

(i) Covtype (j) KDD-Cup-99 (k) Mammography (l) Shuttle

Fig. 1. The total number of true anomalies seen vs. the number of queries for all datasets. Total number of queries for the **smaller** datasets (*top two rows*) is 60. Total number of queries for the **larger** datasets (*bottom row*) is 100.

2) Semi-supervised Anomaly Detector (SSAD): The algorithm proposed by [8] encodes labeled anomalies and nominals as constraints and does not need any labeled data to initialize. The best performing query strategy for SSAD was demonstrated to be the combination strategy called *margin and cluster* [8]. In our experiments we refer to SSAD using this strategy as SSAD-MC. For comparison, we also include in our experiments the query strategy that selects the top ranked anomaly instance for feedback (SSAD-Top).

3) AI2: AI^2 [9] is a system that incorporates analyst feedback for detecting malicious attacks on information systems. It is comprised of an ensemble of unsupervised outlier detection methods and a supervised learning algorithm. In our implementation of AI^2 we use Loda's random projections as the unsupervised ensemble members, and for the supervised algorithm we use L1-regularized Logistic Regression. In our experiments we use a batch size of 2, meaning AI^2 queries the user with one instance suggested by the ensemble of unsupervised algorithms and an instance suggested by the supervised algorithm (if such an instance is available).

4) ATGP: Grill and Pevny [16] propose a supervised algorithm that is also based on maximizing accuracy at the top. Unlike our approach, ATGP does not add pair-wise constraints and it uses a simple gradient approach for inferring the detector weights.

In our experiments, we used the *Mammography* [19] dataset as well as seven datasets from the UCI repository [20]: *Abalone*, *Cardiotocography*, *Thyroid (ANN-Thyroid)*, *Forest Cover (Covtype)*, *KDD-Cup-99*, *Shuttle* and *Yeast*. The number of true anomalies and true nominals in each dataset are shown in Table I. Our implementation of all algorithms and the datasets in our experiments are available online[1]. For SSAD we used the code made available by the authors[2].

[1](https://github.com/shubhomoydas/aad.git)
[2](https://github.com/nicococo/tilitools)

Algorithm 1: Active Anomaly Discovery (AAD)

Data: Raw data $\mathbf{D}$, negative-log-pdfs $\mathbf{H}$, budget B
Initialize the weights $\mathbf{w}^{(0)} = \{\frac{1}{\sqrt{m}}, ..., \frac{1}{\sqrt{m}}\}$
Set $t = 0$, $\mathbf{H}_A = \emptyset$, $\mathbf{H}_N = \emptyset$
while $t \leq B$ **do**
  $t = t + 1$
  Set $\mathbf{a} = \mathbf{H} \cdot \mathbf{w}$ (i.e., $\mathbf{a}$ is the vector of anomaly scores)
  Let $\mathbf{x}_i$ = instance with highest anomaly score (where $i = \arg\max_i(a_i)$)
  Let $\mathbf{z}_i$ = negative log probability vector corresponding to $\mathbf{x}_i$
  Get feedback $\{`anomaly'/`nominal'\}$ on $\mathbf{x}_i$
  if $\mathbf{x}_i$ *is anomaly* **then**
    $\mathbf{H}_A = \{\mathbf{z}_i\} \cup \mathbf{H}_A$
  else
    $\mathbf{H}_N = \{\mathbf{z}_i\} \cup \mathbf{H}_N$
  end
  $\mathbf{w}^{(t)}$ = compute new weights; normalize $\|\mathbf{w}^{(t)}\| = 1$
end

This implementation requires an $n \times n$ kernel matrix and therefore does not scale to large datasets. Therefore, for the larger datasets (*Covtype*, *KDD-Cup-99*, *Mammography*, *Shuttle*) we also include results from a smaller version of the original dataset which was created by sub-sampling 2000 data instances and keeping the ratio of anomalies to nominals roughly the same as in the original dataset. These sub-sampled datasets are named **-sub* (Table I). For the *Cardiotocography* dataset we retained all instances from the *nominal* class as in the original dataset, but down-sampled the *anomaly* instances so that they represent only around 2% of the total data. The rest of the datasets were used in their entirety by all algorithms. SSAD is deterministic and was therefore run once per dataset. Since AAD and AI^2 are randomized, their results were averaged across 10 runs for each dataset. The 95% confidence intervals from these 10 runs are plotted on the curves as error bars. In Table I, the number of dimensions of each dataset (under the 'Dim' column) includes the additional dimensions added after performing the standard transformation of multi-class categorical variables into multiple binary variables using dummy coding.

For AAD, we set the parameters $\tau = 0.03$, $C_A = 100$, and $C_\xi = 1000$ by default for all datasets. AAD is not sensitive to τ over a wide range of values $[0.01, 0.10]$, or to C_A. It generally performs well for $C_\xi > 1$.

Instead of tuning the parameters C_u and C_n in *SSAD*, we give it an advantage by reporting only the best results across all combinations of values $\in \{0.01, 0.1, 1.0, 10.0, 100.0\}$, chosen *after* running the experiments. We fixed $\kappa = 1$ as was set in [8] and used the RBF kernel.

The top two rows in Figure 1 compare the performance of all algorithms on the smaller datasets and the sub-sampled datasets. The bottom row in Figure 1 compares the performance of all algorithms except SSAD on the full versions of the larger datasets. These results show that the AAD curve is consistently one of the best performers across all datasets and it always performs better than its baseline. Although AAD and ATGP are similar algorithms, ATGP performs worse than AAD on most datasets. Our hypothesis is that even though both AAD and ATGP are non-convex and therefore prone to local optima, the pair-wise constraints likely restrict AAD to better parts of the parameter space.

Shuttle is the easiest dataset having the highest percentage of anomalies discovered. For this dataset, almost all instances ranked at the top by the baseline Loda are true anomalies. Therefore, querying the top ranked instance is very effective for AAD and AI^2. Surprisingly, SSAD-Top, which also queries instances ranked at the top by SSAD, discovers fewer anomalies than SSAD-MC. We hypothesize that this might be due to the presence of clustered anomalies in the *Shuttle* dataset, which SSAD-MC is able to exploit more than AAD.

AI^2 performs better than AAD on *Covtype-sub* and when the number of queries is near the end of the budget on *Yeast*. AI^2 performs poorly on the full *Covtype* dataset, which is nearly 100 times larger than *Covtype-sub*. The unsupervised and supervised parts of AI^2 struggle initially to find true anomalies in this much larger dataset, resulting in poor performance with a relatively small budget of 100.

V. Discussion

AAD is based on the intuitive concept that scores assigned to labeled anomalies should be in the top τ-quantile of all scores while scores assigned to nominals should be below. It makes efficient use of analyst feedback to fine-tune the behavior of a reasonably good unsupervised anomaly detector.

Most commonly reported performance metrics like *area under the ROC curve (AUC)*, *average precision (APR)*, and *lift* are computed over the entire (often large) dataset and assume that the analyst is willing to investigate every instance. *Precision at top k (precision@k)* can be used to evaluate the internal ranked list of anomalies after B queries are expended but it fails to account for wasted effort by the analyst when investigating (and labeling) false positives during the data exploration loop. In contrast, the goal of AAD is to maximize the number of true anomalies inspected by the analyst over the B queries.

When the number of false positives among reported anomalies is high, it often discourages users from even getting started with initial exploration. This also causes severe class imbalance when supervised techniques are involved (e.g. [21], [9]). In the extreme case, supervised algorithms are likely to face a *cold start* problem where there are no anomalies in the initial labeled set. In contrast, AAD can usefully incorporate expert feedback even when that feedback only consists of labels for nominal instances. This behavior is due to the fact that AAD relies on an internal ranking model and it attempts to push nominal instances below the top τ-th quantile. AAD can also be used to quickly generate the initial labeled set and bootstrap the supervised methods.

TABLE I
DATASETS

Dataset	Nominal Class	Anomaly Class	Total	Dims	# anomalies(%)
Abalone	8, 9, 10	3, 21	1920	9	29 (1.5%)
ANN-Thyroid-1v3	3	1	3251	21	73 (2.25%)
Cardiotocography	1 (Normal)	3 (Pathological)	1700	22	45 (2.65%)
Covtype	2	4	286048	54	2747 (0.9%)
Covtype-sub	2	4	2000	54	19 (0.95%)
KDD-Cup-99	*'normal'*	*'u2r', 'probe'*	63009	91	2416 (3.83%)
KDD-Cup-99-sub	*'normal'*	*'u2r', 'probe'*	2000	91	77 (3.85%)
Mammography	-1	+1	11183	6	260 (2.32%)
Mammography-sub	-1	+1	2000	6	46 (2.3%)
Shuttle	1	2, 3, 5, 6, 7	12345	9	867 (7.02%)
Shuttle-sub	1	2, 3, 5, 6, 7	2000	9	140 (7.0%)
Yeast	*'CYT'*, *'NUC'*, *'MIT'*	*'ERL'*, *'POX'*, *'VAC'*	1191	8	55 (4.6%)

In future work, we will investigate four directions with AAD. First, we would like to speed up the running time by scaling up the constraint optimization problem in Equation 5. Second, we would like to solve for the values of q_τ and $\mathbf{w}$ simultaneously rather than approximating them by solving them separately. Third, we will explore applying AAD to ensembles of heterogeneous detectors (e.g. [21], [9]) as an alternative to having each ensemble member focus on a particular subspace. Finally, we will explore query strategies that can look ahead further than the current myopic approach (i.e. querying the current most anomalous instance).

VI. Conclusion

We have introduced AAD, which is a simple algorithm that helps an analyst quickly discover anomalies in a dataset. AAD maintains an internal ranking of data instances according to their anomaly scores as computed by the LODA algorithm, which is an ensemble approach to anomaly detection. When AAD receives labeled instances, it adjusts the weights of its ensemble members such that false positives get pushed lower in its internal ranking model while true anomalies bubble up with each labeled instance provided by the expert. Our experiments show that AAD is consistently one of the best performing algorithms.

References

[1] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proceedings of the 24th International Conference on Very Large Data Bases*, 1998, pp. 392–403.

[2] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM Press, 2000, pp. 93–104.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[4] Tom Pevny, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2015.

[5] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[6] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman, "Aladin: Active learning of anomalies to detect intrusions," *Technique Report. Microsoft Network Security Redmond, WA*, vol. 98052, 2008.

[7] N. Nissim, A. Cohen, R. Moskovitch, A. Shabtai, M. Edry, O. Bar-Ad, and Y. Elovici, "Alpd: Active learning framework for enhancing the detection of malicious pdf files," in *Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference*, 2014, pp. 91–98.

[8] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," pp. 235–262, 2013.

[9] K. Veeramachaneni, I. Arnaldo, A. Cuesta-Infante, V. Korrapati, C. Bassias, and K. Li, "Ai2: Training a big data machine to defend," *IEEE International Conference on Big Data Security*, 2016.

[10] D. Pelleg and A. W. Moore, "Active learning for anomaly and rare-category detection," in *Advances in Neural Information Processing Systems*, 2004, pp. 1073–1080.

[11] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[12] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 133–142.

[13] P. Donmez and J. G. Carbonell, "Optimizing estimated loss reduction for active sampling in rank learning," in *Proceedings of the 25th International Conference on Machine Learning*. ACM, 2008, pp. 248–255.

[14] S. Boyd, M. Mohri, C. Cortes, and A. Radovanovic, "Accuracy at the top," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 953–961.

[15] P. Kar, H. Narasimhan, and P. Jain, "Surrogate functions for maximizing precision at the top," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 189–198.

[16] M. Grill and T. Pevný, "Learning Combination of Anomaly Detectors for Security Domain," *Computer Networks*, 2016, doi:10.1016/j.comnet.2016.05.021.

[17] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the Eleventh ACM SIGKDD International Conference on K nowledge Discovery in Data Mining*. New York, NY: ACM, 2005, pp. 157–166.

[18] R. Koenker, *Quantile Regression (Econometric Society Monographs)*. Cambridge University Press, 2005.

[19] K. S. Woods, C. C. Doss, K. W. Bowyer, J. L. Solka, C. E. Priebe, and W. P. Kegelmeyer, "Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 06, pp. 1417–1436, 1993.

[20] "UC Irvine Machine Learning Repository," http://http://archive.ics.uci.edu/ml/, 2007.

[21] T. E. Senator, H. G. Goldberg, A. Memory, W. T. Young, B. Rees, R. Pierce, D. Huang, M. Reardon, D. A. Bader, E. Chow, I. Essa, J. Jones, V. Bettadapura, D. H. Chau, O. Green, O. Kaya, A. Zakrzewska, E. Briscoe, R. I. L. Mappus, R. McColl, L. Weiss, T. G. Dietterich, A. Fern, W.-K. Wong, S. Das, A. Emmott, J. Irvine, J.-Y. Lee, D. Koutra, C. Faloutsos, D. Corkill, L. Friedland, A. Gentzel, and D. Jensen, "Detecting insider threats in a real corporate database of computer usage activity," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1393–1401.