Semi-supervised Learning from Active Noisy Soft Labels for Anomaly Detection

Timo Martens(✉), Lorenzo Perini, and Jesse Davis

DTAI Research Group and Leuven.AI, KULeuven, Leuven, Belgium
{timo.martens,lorenzo.perini,jesse.davis}@kuleuven.be

Abstract. Anomaly detection aims at detecting examples that do not conform to normal behavior. Increasingly, anomaly detection is being approached from a semi-supervised perspective where active learning is employed to acquire a small number of strategically selected labels. However, because anomalies are not always well-understood events, the user may be uncertain about how to label certain instances. Thus, one can relax this request and allow the user to provide soft labels (i.e., probabilistic labels) that represent their belief that a queried example is anomalous. These labels are naturally noisy due to the user's inherent uncertainty in the label and the fact that people are known to be bad at providing well-calibrated probability instances. To cope with these challenges, we propose to exploit a Gaussian Process to learn from actively acquired soft labels in the context of anomaly detection. This enables leveraging information about nearby examples to smooth out possible noise. Empirically, we compare our proposed approach to several baselines on 21 datasets and show that it outperforms them in the majority of experiments.

Keywords: Anomaly Detection · Probabilistic Labels · Noisy Labels

1 Introduction

Anomaly detection is the task of detecting abnormal behaviour in the data. These unexpected occurrences are usually related to critical events, such as machine failure [8], intrusion detection [19] or medical applications [31]. Thus, detecting anomalies in time allows us to save money, preserve privacy and save lives.

Because anomalies are, by definition, rare events, obtaining labels (especially anomalous ones) is often expensive, unethical, or simply time-consuming. Hence, anomaly detection is usually tackled from an unsupervised perspective [10,12]. However, it has been shown in the literature that providing limited, but specific labels to the model can have a large impact on its performance [35,45]. Therefore, one can implement active learning strategies to collect labels strategically, such as those in regions where the model has high uncertainty [1,11,24].

However, sometimes it can be challenging to provide a correct label for a given instance. For example, when labeling abnormal water usage, it may happen

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14169, pp. 219–236, 2023.
https://doi.org/10.1007/978-3-031-43412-9_13

that some normal behaviour (e.g., system maintenance) is infrequent and the user presumes it is anomalous and labels it as such [44]. More generally, an instance's label may be ambiguous, and different annotators may label it in different ways (e.g., crowdsourcing). When reconciling these inconsistencies to get a hard decision, selecting the correct label may be a difficult task [21,39]. A solution to this problem is to relax our request by allowing the user to provide a soft label (i.e., a probability). Thus, one asks how likely it is that an instance is anomalous. Previous work has shown that this relaxation increases performance, especially in highly imbalanced data sets [26,43].

Unfortunately, soft labels that reflect the inherent label probability are hard to collect [9,15]. For example, a user may be overly confident and annotate a slightly excessive usage of water as having a very high probability of being anomalous. Similarly, in crowdsourcing, a group of users may be affected by a biased selection of instances that ends up producing inaccurate probabilities for some specific instances [25]. Thus, asking for a user to provide soft labels often results in examples that are annotated with noisy probabilities. This can have a negative effect on the detector's performance as using incorrect soft labels at training time affects its ability to make accurate predictions at test time. For example, overly high (low) probabilities would make the model sensitive to producing false positives (negatives). Therefore, accounting for the (possible) noise both during training and inference is an important problem.

Additionally, we require a method that has both an unsupervised and supervised component. Many, but not all, anomalies are non-repetitive events. These anomalies are best detected by unsupervised anomaly detectors. However, these unsupervised detectors have difficulties detecting anomalies that look similar to normal instances or might detect some normal behavior as anomalous. Labels can help distinguish these last two cases. Thus, we want to make predictions such that (1) we fall back to unsupervised scores if instances are distant from labeled training data and (2) the instances that are closer to the labeled data receive a score that is mostly based on the soft labels.

Therefore, we fill this gap in the literature by proposing SLADE (Soft Label Anomaly DEtector), the first semi-supervised anomaly detector that learns from noisy soft labels using active learning. Initially, it uses an unsupervised anomaly detector as an indication of how anomalous instances are (prior knowledge). Then, it sets up an active learning loop that (1) measures the uncertainty inherent to dealing with noisy soft labels, (2) uses the uncertainty metric to collect noisy soft labels, and (3) learns from such labels by training a Gaussian Process to model the deviation between the given soft labels and the unsupervised scores. Finally, at inference time, SLADE removes the noise from the soft labels by averaging out the GP's prediction over a Gaussian surface. By summing this average with the unsupervised score, SLADE computes the probability that a test instance is anomalous.

2 Background and Notation

We assume a d-dimensional instance space $\mathcal{X} \subseteq \mathbb{R}^d$ and a binary output space $\mathcal{Y} = \{0, 1\}$ where 1 denotes the anomaly class. Moreover, we assume that we are given an unlabeled dataset $U = \{x_i | x_i \in \mathcal{X}\}_{i=1}^N$ of size N, an initially empty (soft) labeled dataset L, and a label budget $B \in \mathbb{N}$ that indicates how many (soft) labels the user is willing to provide. We now review the necessary background on anomaly detection and Gaussian processes.

2.1 Anomaly Detection

In unsupervised anomaly detection, the goal is to learn a function $s : \mathcal{X} \rightarrow \mathbb{R}$ that assigns real-valued anomaly scores to any instance in $\mathcal{X}$ where, without loss of generality, we assume that higher scores represent more anomalous instances. Unsupervised detectors are trained by making assumptions about what constitutes an anomaly, which typically results in defining how anomalies are dissimilar to normal instances. For example, Isolation Forest (IForest) [22] assumes that anomalies can be easily isolated when randomly splitting the instance space, and assigns anomaly scores inversely proportional to the number of splits needed to isolate an instance. The k-NN outlier detector (kNNO) [2] assumes that anomalies are far away from normals with respect to some notion of distance, and uses the distance to the k-th nearest neighbor as the anomaly score.

A practical issue is how to convert an anomaly score into a hard prediction [32]. One way to do this is to use the contamination factor $\gamma \in [0, 1]$, which is the fraction of anomalies in a dataset [33,34]. Using γ one can define a threshold λ so that a fraction γ of the training data receives an anomaly score greater than λ. For an unseen test instance x_t,

$$y(x_t) = \begin{cases} 0 & s(x_t) \leq \lambda \\ 1 & s(x_t) > \lambda \, . \end{cases} \tag{1}$$

Recently, there is increasing recognition that incorporating strategically chosen labeled instances is important for improving the performance of anomaly detectors [35,45]. Active learning (AL) is commonly used to select which instances to label [17,41]. At a high level, it is possible to distinguish among three approaches to AL [24]: *uncertainty-based* strategies aim to select the unlabeled data samples with the highest uncertainty [11], *diversity-based* strategies aim to maximize the diversity among the labeled training data [1] and *combined* strategies integrate the advantages of these two [6]. The first category is widely used due to its simplicity and strong performance. Starting with an unlabeled dataset U and an empty (soft) labeled dataset L, a detector is learned in an unsupervised manner. Then, the following steps are repeated until a given label budget is exhausted. First, query a human annotator to provide a (soft) label for the strategically chosen instances. In uncertainty sampling, one approach is to use the probabilistic gap $|P(Y = 1|x) - P(Y = 0|x)|$ where smaller gaps indicate higher uncertainty. Second, the queried instances and their (soft) labels are added to L and the model is retrained using this newly expanded dataset.

2.2 Gaussian Processes

A Gaussian process (GP) is a collection of random variables over the instance space, such that any finite subset of them have a joint Gaussian distribution [37]. Roughly speaking, a GP can be seen as a distribution over functions $f\colon \mathcal{X} \to \mathbb{R}$ such that for any $x, x' \in \mathcal{X}$

$$f(x) \sim \mathcal{GP}(m(x), \mathcal{K}(x, x')),$$

where $m\colon \mathcal{X} \to \mathbb{R}$ is called the mean function, and $\mathcal{K}\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the covariance function (otherwise known as the kernel). The Gaussian process is completely characterized by these two functions m and $\mathcal{K}$, which define

$$\mathbb{E}[f(x)] = m(x) \quad \text{and} \quad \textsc{Cov}[f(x), f(x')] = \mathcal{K}(x, x').$$

Picking an appropriate prior mean and kernel enables encoding prior beliefs of the data-generating process into the model. More importantly, the GP fully relies on these prior beliefs to make predictions for an unseen instance that falls in a region far from any training instance. Given a training set of pairs $\mathcal{R} = \{(x_i, r_i)\}_{i=1}^{|\mathcal{R}|}$, where $r_i \in \mathbb{R}$, the posterior distribution of a GP for any $x, x' \in \mathcal{X}$ is

$$\begin{aligned} f|\mathcal{R} &\sim \mathcal{GP}(m_{\mathcal{R}}, \mathcal{K}_{\mathcal{R}}) \\ m_{\mathcal{R}}(x) &= m(x) + \Sigma_{x,X} \left(\Sigma_{X,X}\right)^{-1} (\mathbf{r} - m(X)) \\ \mathcal{K}_{\mathcal{R}}(x, x') &= \mathcal{K}(x, x') - \Sigma_{x,X} \left(\Sigma_{X,X}\right)^{-1} \Sigma_{X,x'}\,, \end{aligned} \tag{2}$$

where the elements of $\Sigma_{a,b}$ depend on the kernel $(\Sigma_{a,b})_{i,j} = \mathcal{K}(a_i, b_j)$, which makes $\Sigma_{X,X}$ the training-training covariance matrix, and $\Sigma_{x,X}$, $\Sigma_{X,x'}$, respectively, $1 \times |\mathcal{R}|$ and $|\mathcal{R}| \times 1$ covariance vectors. Note that the posterior covariance is always lower than the prior due to the subtraction of a strictly positive term.

Given a test set $T = \{x_t\}_{t=1}^{|T|}$, the GP predicts a posterior multivariate normal distribution ($|T|$-dimensional) $\mathcal{N}(m_{\mathcal{R}}(T), \mathcal{K}_{\mathcal{R}}(T, T))$. Note, that each individual instance has a Gaussian marginal distribution that can be used for instance-wise predictions. In practice, one can derive the final prediction from the given distribution by either taking a sample (Bayesian perspective) or extracting the mean (frequentist perspective). In this work, we use the latter.

3 SLADE

Our goal is to learn a model to estimate the probability that an instance is anomalous in an active learning setting where a user provides soft labels. Starting from an unlabeled dataset $U = \{x_n | x_n \in \mathcal{X}\}_{n=1}^{N}$, an empty soft labeled dataset L, and a label budget B, the algorithm can iteratively query instance $x \in U$. However, instead of receiving its exact label, the user provides a real value $p \in [0, 1]$ indicating the probability that the instance belongs to the anomaly class.

Designing an approach to learn in this setting has three key challenges. First, we need an informative unsupervised score about what is and is not likely to be anomalous. This allows the model to output probabilities even in regions where no soft labels are given. Second, we need a way to combine the weak supervision provided by the soft labels with this unsupervised score such that (1) we fall back to the initial scores if instances are distant from labeled training data and (2) the instances that are closer to the soft labeled data in L receive a score that is mostly based on those labels. Third, we need to explicitly model the uncertainty that is inherent when working with soft labels.

We address these challenges by combining unsupervised anomaly detection with a Gaussian process. Intuitively, the anomaly detector will provide an informative prior for the GP. A key question is what the GP should model. One choice would be to have it directly model the soft labels. However, because the labels are uncertain and noisy, we want to decouple the noise arising from the soft labels and the uncertainty of unsupervised scores. Therefore, we model the deviation of the soft labels from the unsupervised prior. When making a prediction, we propose a novel way to combine the estimated deviation and the unsupervised score in a noise-robust way. Next, we describe our training and inference procedures in more detail.

3.1 Training

SLADe constructs the informative prior by taking a completely unsupervised approach. First, SLADe trains an unsupervised anomaly detector on U that can compute an anomaly score for any instance $x \in \mathcal{X}$, which is denoted as $s(x)$. SLADe is detector agnostic and we will discuss possible choices in the experimental evaluation. Second, we want to learn the deviation of the soft labels from these scores. However, working with the raw scores is not possible because scores provided by different unsupervised models have different meanings. Moreover, anomaly scores often cannot be interpreted as probabilities (e.g., kNNo assigns a distance) and thus, in this form they can not be compared with soft labels (i.e., probabilities). Therefore, we apply the linear unification transformation (i.e., min-max normalisation) [18]

$$\tilde{s}(x) = \frac{s(x) - min(\mathbf{s})}{max(\mathbf{s}) - min(\mathbf{s})}$$

to map anomaly scores into $[0, 1]$, where $\mathbf{s} = \{s_1, \ldots, s_N\}$ are the anomaly scores for U. We opt for linear unification because we do not want to introduce strong assumptions on the unsupervised scores (which, working as a prior, is supposed to be flexible [46]).

Our GP models the deviation between the user-provided soft labels and these prior probabilities and it is initialized as $g_0 \sim \mathcal{GP}(0, \mathcal{K})$. The posterior GP is then defined as

$$g_0 | L_0 \sim \mathcal{GP}(m_{L_0}, \mathcal{K}_{L_0}),$$

where $L_0 = \{(x_j, p_j - \tilde{s}(x_j))\colon (x_j, p_j) \in L\}$ denotes a dataset containing the difference between the soft labels (i.e., p_j) and the unified unsupervised scores of the training data in L. To gather soft labeled training data and train the GP, we run an active learning loop. Given a label budget B, we repeat the following steps until our label budget is exhausted. (1) We query the instance $x_* \in U$ where the model is the most uncertain. Quantifying uncertainty requires assigning a prediction to each instance in U. By combining the unsupervised prior $\tilde{s}$ with the GP's mean m_{L_0}, we obtain a first probability estimate:

$$P_1(Y = 1|x, L) = \tilde{s}(x) + m_{L_0}(x)\,. \tag{3}$$

Model uncertainty can arise for two reasons: making weak predictions (≈ 0.5) and a lack of labeled instances in certain regions of the instance space. To capture both types of uncertainty, we use Kapoor et al. [16]'s strategy to query labels for

$$\underset{x_* \in U}{argmin} \frac{|0.5 - P_1(Y = 1|x_*, L)|}{\sqrt{\mathcal{K}_{L_0}(x_*, x_*)}}\,.$$

This formula assigns low scores if (a) the posterior probability is close to 0.5 (small numerator), or (b) if the instance is far from the labeled instances and hence has high prediction variance (big denominator). (2) Finally, SLADE updates $L = L \cup \{(x_*, p_*)\}$ and $U = U \setminus \{x_*\}$. Subsequently, $g_0|L_0$ is updated with the newly obtained soft labels.

3.2 Inference

Given an unseen test instance x_t and a set of soft labels L, computing the posterior probability $P(Y = 1|x_t, L)$ is challenging for the following reason. An initial estimate of the posterior probability can be obtained via Eq. 3. However, this probability is heavily affected by noisy soft labels. Per definition, the GP predicts the exact soft labels for each soft-labeled training instance. Consequently, if x_t is in close proximity to a noisy soft label, the predicted posterior probability would be affected by this noise.

We propose to mitigate the effect of noisy labels as follows. We distinguish between two types of test instances: (1) those that are far from the training data and (2) those that have many training instances nearby. Since the unsupervised anomaly scores model the proximity to other data points, we can use this as a measure without introducing any new assumptions (i.e. high anomaly scores represent distant instances). For the first type of test instances, there is no reason to try and fix the noise. They are far from the training data and will thus not be influenced by noise. The second type, on the other hand, is influenced by label noise. We cope with this problem by smoothing out the estimated deviation over a Gaussian surface that has x_t as the center and a given variance σ_t^2. Formally,

$$P_2(Y = 1|x_t, L) = \tilde{s}(x_t) + \mathbb{E}_{V \sim \mathcal{N}(x_t, \sigma_t^2)}[m_{L_0}(V)], \tag{4}$$

where V is a normally distributed random variable. Using the surrounding instances forces the model to use more soft labels when computing the posterior probability, which clearly averages out the negative effects that the presence of noise has on the model. σ_t is dependent on x_t and we define it as one-third of the radius of a hypersphere with center x_t that captures $q\%$ of the instances in U. Thus, for every test instance, we average out over the same number of training data. We then formalize our final probability estimate as

$$\hat{P}(Y=1|x_t,L) = \begin{cases} P_1(Y=1|x_t,L) & s(x_t) > \lambda \\ P_2(Y=1|x_t,L) & s(x_t) \leq \lambda \,, \end{cases} \tag{5}$$

where λ denotes the anomaly score threshold as defined in Eq. 1. A hard prediction is obtained by setting a threshold, typically 0.5, on the probability estimates.

4 Experiments

We address the following two research questions: **Q1:** How do the methods compare under various noise regimes? **Q2:** How sensititive is SLADe to the choice of its hyperparameters?

4.1 Experimental Setup

Methods. We compare SLADe[1] against four baselines. Conceptually, these can be divided into two groups. The first group learns directly from probabilistic labels: GP [31] simply uses a Gaussian Process to model the soft labels without including the unsupervised prior, while P-SVM [20] uses a Support Vector Machine (SVM) with class labels that are weighted by the given soft labels. The second group cannot operate directly on the soft labels. Therefore, we convert them to hard labels by flipping a weighted coin. Then we apply traditional semi-supervised models. SSDO [44] is a propagation-based detector that uses the distance to hard labels to assign anomaly scores. HIF [23] is a semi-supervised variant of the widely used unsupervised Isolation Forest [22] that improves its anomaly scores by adding the distance to the anomalous hard labels.

Data. We evaluate our method and the baselines on 21 benchmark datasets that are widely used in the anomaly detection literature [4,12]. These datasets vary in size, number of features, and proportion of anomalies. To limit the computational cost of the experiments, we subsample each dataset to at most 5000 instances keeping the same proportion between normals and anomalies. See Table 1 for the characteristics of the datasets.

[1] The code and Supplement are available via https://github.com/TimoM99/SLADe.

Table 1. Characteristics (full size, subsampled size, number of features d, contamination factor γ) of the 21 benchmark datasets used for the experiments.

Dataset	Full size	Size	d	γ	**Dataset**	Full size	Size	d	γ
ALOI	50,000	5000	27	0.030	Pen	9,868	5000	16	0.002
Annthy	7,200	5000	21	0.075	Pima	555	555	8	0.099
Arrhy	271	271	259	0.100	Shuttle	1,013	1013	9	0.013
Cardio	2,112	2112	21	0.221	Spam	2,661	2661	57	0.050
Glass	213	213	7	0.042	Stamps	340	340	9	0.091
Heart	166	166	13	0.096	Wave	3,443	3443	21	0.029
Hepa	80	80	19	0.163	WBC	223	223	9	0.045
Iono	350	350	32	0.357	WDBC	367	367	30	0.027
KDD	48,113	5000	40	0.040	Wilt	4,819	4819	5	0.053
Page	5,393	5000	10	0.095	WPBC	198	198	33	0.237
Parkin	60	60	22	0.200					

Setup. Our setup can be divided into three parts: (1) generating the ground-truth soft labels, (2) introducing the noise, and (3) evaluating the methods.

The first part requires modeling the human annotator: given an instance x, a soft label p indicates the proportion of anomalous labels that we would obtain if we queried x multiple times. Moreover, similar instances are likely to obtain similar probabilities. We model this aspect by training a Random Forest with low depth ($= 4$) on the original dataset and use it to compute the soft labels as class probabilities. The low depth guarantees that Random Forest does not push all probabilities to the extremes (0 or 1) but assigns smooth values over $[0, 1]$.

In the second part, we introduce noise into the soft labels. We use a standard transformation [7] that changes the label p into $1-p$ for a fixed percentage of the soft labels. The noisy instances are picked uniformly at random. The percentage of swapped labels is the noise level of the dataset.

Finally, for each of the 21 datasets, we run the following experiment: (i) We randomly split the dataset into 80% training and 20% test set; (ii) We compute the ground-truth soft labels and add the given level of noise to the training soft labels; (iii) We run the active learning loop with a label budget $B = 60\%$ of the training set size N, which we split into 12 rounds of 5% each. We choose a label budget of 60% for completeness reasons. All baseline methods also employ uncertainty sampling. (iv) We evaluate the Area Under the Receiving Operating Curve (AUROC) [14] of each method at every iteration of the loop. As the test set also has soft labels, we sample a hard label to make the evaluation consistent within our probabilistic setting. To average out the randomness introduced by sampling labels, we repeat the active learning loop 20 times. All four steps are then repeated five times. We carry out a total of $5 \times 20 \times 21 = 2100$ experiments.

Hyperparameters. SLADe has three hyperparameters. We choose IForest [22] as the unsupervised method. We use the Matèrn kernel with $\nu = \frac{1}{2}$

in the GP as it is widely used in the literature [36]. Moreover, we optimize the length scale hyperparameter of the Gaussian Process by maximizing the log marginal likelihood [37]. Finally, we set $q = 2$. SSDO uses the same prior model as SLADe and the default values for α and k. HIF has two hyperparameters: α_1 and α_2. Since the paper does not suggest any values, we set both to 0.5, which makes a fair weighting between the different parts of the score. P-SVM utilizes an RBF kernel with the default parameters [20]. Finally, GP relies on a Gaussian Process that has the same hyperparameters as for SLADe.

4.2 Experimental Results

Q1. Comparing the Methods. We want to evaluate SLADe on two aspects: (1) its robustness against noise and (2) its ability to rank anomalies. Therefore, we compare SLADe against the baselines on three different noise levels and compare both their noise-robustness and performance at different label percentages.

First, we compare SLADe against the baselines for each label frequency of the active learning loop under the three noise levels (0%, 10%, 20%). For this task, we plot the learning curve, which has on the x-axis the label percentage as a proportion of the dataset's size, and, on the y-axis, the methods' AUROC. Figure 1 shows the results on five representative datasets, while the Supplement includes the plots for all the remaining datasets. Regardless of the noise, SLADe clearly outperforms all the baselines on Shuttle (left plot), while it performs similarly to the baselines on Pima and Heart (second and third plots). On the other hand, on Page and Iono (right plots), SLADe obtains competitive AUROC values with no noise present while outperforming all the baselines at higher noise levels (10% and 20%). Overall, the major strength of SLADe is the ability to improve its performance when acquiring (possibly noisy) soft labels: on Shuttle, SLADe's learning curve is steeper than all the baselines' for all noise levels. On the other hand, looking at Page and Iono, all methods' learning curves are flat, but SLADe's does not deteriorate as hard as the baselines when introducing higher noise levels.

Second, we dive deeper into the noise-robustness of the methods. Therefore, we aggregate the results on a per-dataset basis and measure how their performance decreases when moving from a setting with no noise to a setting with (a) 10% and (b) 20% of noise. Figure 2 reports the methods' mean AUROC drop aggregated over all of the label percentages for the two scenarios. The star (cross) markers indicate the mean AUROC with no noise (the given level of noise), while the length of the segment is indicative of how robust each model is against noise: the shorter the segment, the smaller the change of AUROC, and the more robust the model. The results show that SLADe obtains the lowest/similar (i.e., within a gap of 0.01) drop in performance in 13 out of 21 datasets when the noise goes from 0% to 10%, while it does so on six datasets when increasing the noise to 20%. Unsurprisingly, the second-best baseline is HIF, which is naturally noise-robust because it only leverages anomalous labels to assign scores, which hides the negative effect of noisy negative labels provided by the user. In fact, HIF obtains the lowest drop in performance on six datasets under 10% noise, and

Table 2. Wins (W), Draws (D), and Losses (L) of **SLADe against each baseline** in terms of average AUROC per dataset, for each label percentage, under 20% of noise. A draw means that the absolute difference in AUROC is ≤ 0.01.

	SSDO			**P-SVM**			**HIF**			**GP**		
Labels	W	D	L	W	D	L	W	D	L	W	D	L
5%	**13**	5	3	**16**	1	4	**10**	5	6	**17**	2	2
10%	**15**	3	3	**16**	1	4	**12**	5	4	**16**	2	3
15%	**17**	3	1	**17**	0	4	**12**	8	1	**15**	2	4
20%	**16**	5	0	**17**	0	4	**17**	4	0	**15**	0	6
25%	**17**	4	0	**18**	0	3	**15**	6	0	**15**	0	6
30%	**14**	6	1	**18**	0	3	**18**	3	0	**14**	2	5
35%	**13**	5	3	**17**	1	3	**17**	3	1	**15**	1	5
40%	**13**	6	2	**17**	1	3	**17**	2	2	**15**	1	5
45%	**12**	6	3	**17**	1	3	**17**	2	2	**14**	3	4
50%	**12**	4	5	**17**	0	4	**17**	2	2	**14**	2	5
55%	**12**	3	6	**17**	1	3	**17**	2	2	**14**	2	5
60%	**12**	3	6	**16**	2	3	**16**	2	3	**11**	6	4

nine datasets under 20% noise. Furthermore, GP is the most affected by the noise: because it only learns from the given soft labels, incorrect probabilities have a strong impact on the surrounding test instances.

Finally, because our task is to develop a noise-resistant model, we zoom in on the high noise scenario (20%) and analyze how often SLADe outperforms each baseline.[2] Table 2 shows the number of times (out of 21) SLADe's average AUROC is higher (Win), within a margin of 0.01 (Draw) or lower (Loss) than that of the baselines at every label percentage. For any label percentage SLADe never loses more than six times against any baseline. As expected, SLADe outperforms HIF more often at higher label percentages because HIF only uses positive labels. Moreover, against GP, SLADe wins more in the lower label percentage settings (which are more realistic in Active Learning) because SLADe needs less data to learn effectively.

Q2. Sensitivity Analysis. We evaluate the effect of varying SLADe's three hyperparameters: the unsupervised anomaly detector, the GP's kernel, and the percentage of training instances inside the hypersphere, q, used to fix the noise at inference time. We assume a default level of noise equal to 10% and vary one hyperparameter at a time while keeping the other two as specified in Sect. 4.1. We subsample the datasets to at most 500 instances for computational reasons.

Table 3 shows SLADe's AUROC averaged over all datasets for different label percentages when using Isolation Forest (IForest) [22], One-Class SVM

[2] Results for 0% and 10% noise are, for completeness, in the Supplement.

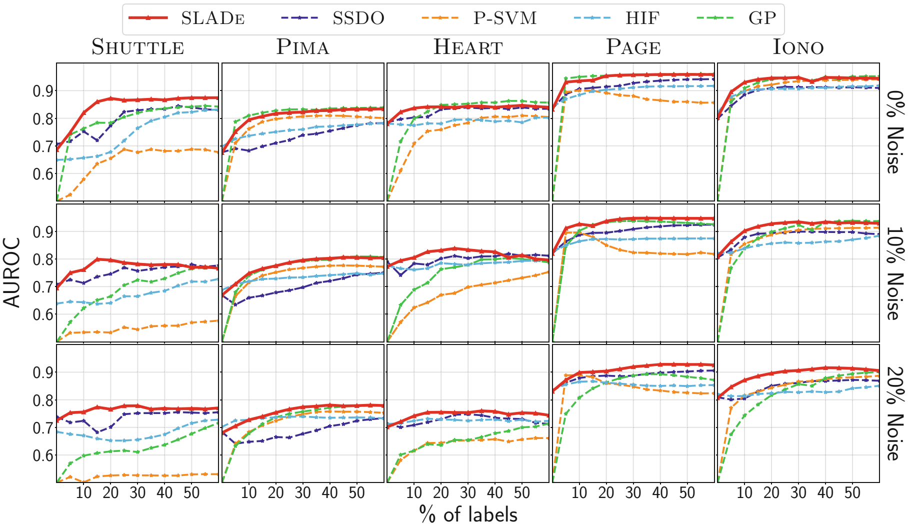

Fig. 1. Learning curves for all methods on five representative datasets for three different noise levels (0%, 10%, 20%). On the x-axis we vary the label percentage, while on the y-axis we report the average AUROC (higher is better).

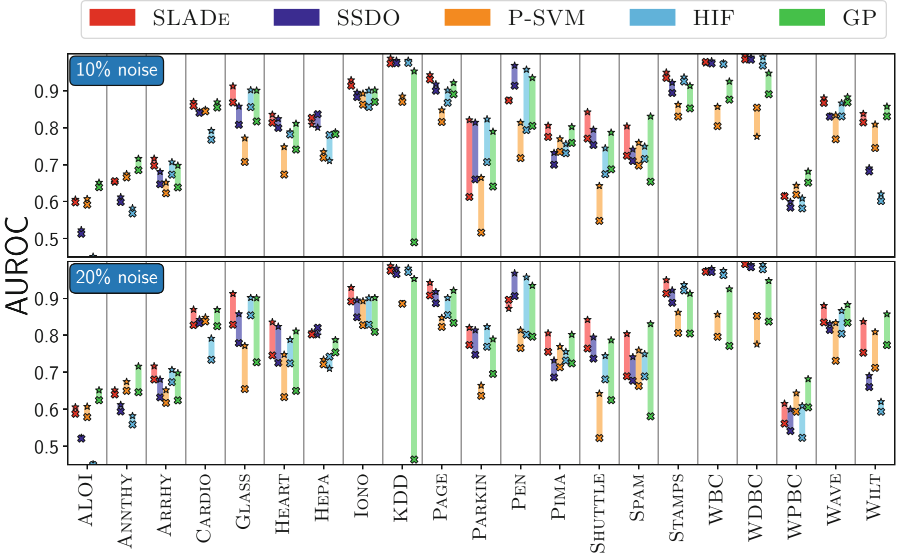

Fig. 2. Comparison on all 21 datasets between the methods' mean AUROC when moving from a clean setting to 10% (top) and 20% (bottom) of noise. The AUROC is aggregated over all percentages of labels. For every dataset and method, the star/cross marker indicates the AUROC with no noise/given level of noise. The length of the segment quantifies the drop in AUROC when introducing noise (shorter is more resistant).

Table 3. AUROC (avg ± std) of SLADE for different unsupervised detectors.

Labels	**Unsupervised detector**			
	IForest	LOF	kNNO	OCSVM
0%	0.730 ± 0.181	0.669 ± 0.180	0.707 ± 0.173	0.664 ± 0.223
5%	0.745 ± 0.178	0.724 ± 0.175	0.750 ± 0.167	0.725 ± 0.204
10%	0.776 ± 0.174	0.763 ± 0.180	0.787 ± 0.165	0.744 ± 0.201
15%	0.800 ± 0.168	0.780 ± 0.177	0.798 ± 0.166	0.776 ± 0.183
20%	0.817 ± 0.163	0.791 ± 0.174	0.808 ± 0.160	0.794 ± 0.179
25%	0.826 ± 0.160	0.793 ± 0.179	0.816 ± 0.155	0.800 ± 0.173
30%	0.833 ± 0.154	0.805 ± 0.169	0.818 ± 0.153	0.816 ± 0.163
35%	0.839 ± 0.150	0.807 ± 0.161	0.825 ± 0.145	0.821 ± 0.159
40%	0.841 ± 0.148	0.816 ± 0.158	0.830 ± 0.137	0.822 ± 0.159
45%	0.843 ± 0.146	0.817 ± 0.156	0.832 ± 0.136	0.823 ± 0.158
50%	0.843 ± 0.143	0.821 ± 0.148	0.834 ± 0.133	0.828 ± 0.154
55%	0.844 ± 0.141	0.819 ± 0.148	0.835 ± 0.131	0.827 ± 0.152
60%	0.844 ± 0.140	0.819 ± 0.146	0.833 ± 0.134	0.826 ± 0.152

(OCSVM) [42], Local Outlier Factor (LOF) [13] and the k-NN outlier detector (kNNO) [2] as unsupervised detectors to assign the anomaly scores. SLADE seems to be robust to the selected anomaly detector as all approaches perform similarly. There are small differences for the three lowest label budgets, where using IForest offers some performance gains. This happens because IForest assigns better rankings to the anomalies, as confirmed by [12] as well. A bad unsupervised model will thus require a certain number of labels before it is able to accurately detect anomalies. Therefore, selecting the correct unsupervised model is an important decision.

Table 4 shows the AUROC averaged over all datasets for different label percentages when using four variants of the Matérn kernel [36] as the covariance function of the GP. We vary its hyperparameter $\nu \in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, +\infty\}$, where $\nu = +\infty$ represents the Radial Basis Function (RBF) kernel [3]. The results illustrate that SLADE has the highest performance for $\nu = \frac{1}{2}$, in agreement with the existing literature on Gaussian Processes [36]. Unsurprisingly, results show that SLADE's performance deteriorates when increasing the hyperparameter ν: because ν indicates the smoothness of the GP's kernel (i.e., high differentiability), high values of ν underpin the assumption that the class probability function is smooth, which is not true in several real-world datasets. Moreover, the effect of changing ν increases with the number of soft labels, which ends up being > 0.06 against $\nu = +\infty$ with 60% of soft labels.

Table 5 shows the AUROC averaged over all datasets for varying label budgets for $q \in [0.5, 1, 2, 5, 10]$. The results show that the value of this hyperparameter has a negligible impact on SLADE's performance. Therefore, we set

q's default value to 2, as it is an in-between value that avoids averaging over too many instances, which might slightly decrease the performance with little noise, and averaging over almost no instance, which would make the model too sensitive to noise.

Table 4. AUROC (avg ± std) of SLADe for different values of the Matérn kernel's hyperparameter ν.

	Matérn Kernel			
Labels	$\nu = 0.5$	$\nu = 1.5$	$\nu = 2.5$	$\nu = +\infty$
0%	0.728 ± 0.183	0.728 ± 0.183	0.728 ± 0.183	0.728 ± 0.183
5%	0.742 ± 0.182	0.733 ± 0.182	0.727 ± 0.184	0.719 ± 0.186
10%	0.770 ± 0.180	0.758 ± 0.180	0.753 ± 0.179	0.745 ± 0.177
15%	0.794 ± 0.176	0.779 ± 0.175	0.771 ± 0.177	0.759 ± 0.178
20%	0.809 ± 0.175	0.791 ± 0.174	0.783 ± 0.175	0.765 ± 0.178
25%	0.820 ± 0.167	0.798 ± 0.171	0.789 ± 0.174	0.770 ± 0.178
30%	0.827 ± 0.162	0.804 ± 0.168	0.795 ± 0.170	0.774 ± 0.174
35%	0.832 ± 0.157	0.808 ± 0.165	0.798 ± 0.166	0.776 ± 0.174
40%	0.836 ± 0.152	0.812 ± 0.160	0.798 ± 0.166	0.776 ± 0.174
45%	0.837 ± 0.151	0.812 ± 0.159	0.799 ± 0.164	0.776 ± 0.173
50%	0.839 ± 0.147	0.814 ± 0.156	0.799 ± 0.161	0.778 ± 0.171
55%	0.839 ± 0.145	0.813 ± 0.154	0.799 ± 0.159	0.775 ± 0.169
60%	0.841 ± 0.143	0.813 ± 0.152	0.798 ± 0.158	0.775 ± 0.168

5 Related Work

There is, to our knowledge, no work that tackles learning from active noisy soft labels in anomaly detection. However, three related research lines exist that are of interest, of which the first two relate to traditional binary classification tasks.

Learning from Soft Labels. The literature on learning from soft labels consists of three common approaches: ranking methods, regression methods and traditional methods adapted for soft labels. (1) Ranking methods solve a constrained optimization problem where the constraints are pairwise rankings between the soft labels [26,27,38]. (2) Regression methods use soft labels as target values in their learning mechanism [31]. (3) Probabilistic Support Vector Machines (P-SVM) use soft labels to micro-steer the obtained margin [20,28]. Empirical evaluation [26] shows that this third category performs best. However, in Sect. 4.2 we showed that SLADe outperforms P-SVM.

Learning from Noisy Hard Labels. The existing work on models that are designed to be noise-robust mostly takes a supervised approach [5,7,48]. These make strong assumptions that do not hold in our setting. For instance, there is no correctly labeled subset of data available [48]. A strictly weaker assumption is the availability of a large set of noisy data [5]. It is non-trivial how to adapt these methods for small sets of noisy labels.

Weakly Supervised Models. Some existing literature in anomaly detection deals with weak supervision. For example, some semi-supervised methods need access only to a small set of clean labels [29,30,40,47]. However, it is unclear how to extend them to deal with soft labels.

Table 5. AUROC (avg ± std) of SLADE for different values of q (% of training instances inside the hypersphere).

	q				
Labels	0.5	1	2	5	10
5%	0.739 ± 0.183	0.740 ± 0.183	0.740 ± 0.184	0.740 ± 0.184	0.741 ± 0.184
10%	0.769 ± 0.183	0.768 ± 0.184	0.768 ± 0.185	0.767 ± 0.185	0.768 ± 0.185
15%	0.794 ± 0.175	0.793 ± 0.176	0.792 ± 0.177	0.791 ± 0.177	0.790 ± 0.178
20%	0.810 ± 0.169	0.810 ± 0.170	0.809 ± 0.170	0.806 ± 0.172	0.805 ± 0.172
25%	0.821 ± 0.163	0.820 ± 0.163	0.819 ± 0.166	0.816 ± 0.168	0.815 ± 0.167
30%	0.829 ± 0.156	0.828 ± 0.157	0.827 ± 0.159	0.824 ± 0.162	0.823 ± 0.162
35%	0.835 ± 0.152	0.834 ± 0.154	0.833 ± 0.155	0.830 ± 0.157	0.828 ± 0.157
40%	0.836 ± 0.150	0.836 ± 0.151	0.834 ± 0.152	0.832 ± 0.154	0.830 ± 0.155
45%	0.838 ± 0.147	0.837 ± 0.148	0.836 ± 0.149	0.833 ± 0.152	0.831 ± 0.153
50%	0.840 ± 0.144	0.839 ± 0.145	0.838 ± 0.146	0.835 ± 0.149	0.833 ± 0.150
55%	0.840 ± 0.141	0.839 ± 0.142	0.838 ± 0.143	0.836 ± 0.146	0.832 ± 0.148
60%	0.840 ± 0.139	0.840 ± 0.141	0.839 ± 0.142	0.836 ± 0.145	0.833 ± 0.146

6 Conclusion

This paper tackled the challenge of learning a model that estimates the probability of an instance being anomalous in an active learning setting where the user provides noisy soft labels. The soft labels indicate the probability that the instance belongs to the anomaly class. The key challenges were how to (1) have an initial indication of how likely instances are anomalous without having access to labels, (2) combine the obtained soft labels with the initial unsupervised scores, (3) model the uncertainty when learning from soft labels, and (4) develop a noise-robust approach that smooths out the noisy probabilities. We proposed SLADE, the first semi-supervised anomaly detector that leverages the noisy soft labels by (1) computing the anomaly scores using an unsupervised anomaly detector, and (2) fixing the scores by modeling their deviation from the given soft labels through a GP. In the active learning loop, it queries the most informative instances by quantifying the model uncertainty that arises from

(a) receiving weak soft labels (e.g., 0.5) and (b) the lack of labels. Finally, at inference time, it smooths out the noise by averaging the GP prediction over a Gaussian surface with adaptive variance. Experimentally on 21 datasets, we showed that SLADe is noise-robust and that it performs better than several baselines on the majority of cases.

Ethical Statement. In general, any work on anomaly detection is beneficial to society. In many applications, it is important to detect anomalies in due time as they are often related to critical events, such as machine failure [8], intrusion detection [19] or medical applications [31]. Being able to detect anomalies in time, thus allows us to save money, preserve privacy and save lives. However, the use of anomaly detection and soft labels in certain settings raises some ethical concerns that need to be considered. One of the primary concerns is the potential for discrimination against some minorities. As anomaly detection techniques are designed to identify instances that deviate from "normal behavior", it is possible that someone with malicious intentions misuses anomaly detectors to discriminate against specific groups by labeling their behavior as "anomalous". Another due ethical consideration relates to the potential violation of privacy that may result from failing to detect anomalies in particular applications. For example, in intrusion detection, the failure to detect anomalous hacker activity could compromise some people's privacy. Finally, the traditional labeling approaches for anomaly detection usually involve the use of an expert. However, collecting soft labels instead of hard labels allows for the use of multiple cheap labor forces instead of a single domain expert. While this may lower the cost of labeling data, it raises ethical concerns regarding the exploitation of cheap labor and the potential for unfair practices.

Acknowledgment. This work is supported by the FWO-Vlaanderen (aspirant grant 1166222N to LP and G0D8819N to JD and TM) and the Flemish government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme (JD, LP).

References

1. Abe, N., Zadrozny, B., Langford, J.: Outlier detection by active learning. In: 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 504–509. Springer (2006)
2. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS, vol. 2431, pp. 15–27. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45681-3_2
3. Buhmann, M.D.: Radial basis functions. Acta Numer. **9**, 1–38 (2000)
4. Campos, G.O., et al.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining Knowl. Discov. **30**, 891–927 (2016)
5. Ding, Y., Wang, L., Fan, D., Gong, B.: A semi-supervised two-stage approach to learning from noisy labels. In: 2018 IEEE Winter Conference on Applications of Computer Vision, pp. 1215–1224. IEEE (2018)

6. Ebert, S., Fritz, M., Schiele, B.: Ralf: a reinforced active learning formulation for object class recognition. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2012)
7. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE Trans. Neural Netw. Learn. Syst. **25**(5), 845–869 (2013)
8. Fujimaki, R., Yairi, T., Machida, K.: An approach to spacecraft anomaly detection problem using kernel feature space. In: 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 401–410. Association for Computing Machinery (2005)
9. Griffin, D., Tversky, A.: The weighing of evidence and the determinants of confidence. Cognit. Psychol. **24**(3), 411–435 (1992)
10. Guthrie, D., Guthrie, L., Allison, B., Wilks, Y.: Unsupervised anomaly detection. In: 20th International Joint Conference on Artificial Intelligence, pp. 1624–1628. Morgan Kaufmann Publishers (2007)
11. Hacohen, G., Dekel, A., Weinshall, D.: Active learning on a budget: opposite strategies suit high and low budgets. In: 39th International Conference on Machine Learning, pp. 8175–8195. PMLR (2022)
12. Han, S., Hu, X., Huang, H., Jiang, M., Zhao, Y.: Adbench: anomaly detection benchmark. Adv. Neural Inf. Process. Syst. **35**, 32142–32159 (2022)
13. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognit. Lett. **24**(9–10), 1641–1650 (2003)
14. Huang, J., Ling, C.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. Knowl. Data Eng. **17**(3), 299–310 (2005)
15. Juslin, P., Olsson, H., Winman, A.: The calibration issue: theoretical comments on suantak, bolger, and ferrell (1996). Organiz. Behav. Human Decis. Process. **73**(1), 3–26 (1998)
16. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active learning with gaussian processes for object categorization. In: 11th IEEE International Conference on Computer Vision, pp. 1–8. IEEE (2007)
17. Kowalska, K., Peel, L.: Maritime anomaly detection using gaussian process active learning. In: 15th IEEE International Conference on Information Fusion, pp. 1164–1171. IEEE (2012)
18. Kriegel, H.P., Kroger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: 2011 SIAM International Conference on Data Mining, pp. 13–24. SIAM (2011)
19. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J.: A comparative study of anomaly detection schemes in network intrusion detection. In: 2003 SIAM International Conference on Data Mining, pp. 25–36. SIAM (2003)
20. Lin, C.F., Wang, S.D.: Fuzzy support vector machines. IEEE Trans. Neural Netw. **13**(2), 464–471 (2002)
21. Littlestone, N., Warmuth, M.: The weighted majority algorithm. Inf. Comput. **108**(2), 212–261 (1994)
22. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE (2008)
23. Marteau, P.F., Soheily-Khah, S., Béchet, N.: Hybrid isolation forest-application to intrusion detection. arXiv preprint arXiv:1705.03800 (2017)
24. Monarch, R.M.: Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI. Simon and Schuster (2021)
25. Nassar, L., Karray, F.: Overview of the crowdsourcing process. Knowl. Inf. Syst. **60**, 1–24 (2019)

26. Nguyen, Q., Valizadegan, H., Hauskrecht, M.: Learning classification models with soft-label information. J. Am. Med. Inf. Assoc. **21**(3), 501–508 (2014)
27. Nguyen, Q., Valizadegan, H., Seybert, A., Hauskrecht, M.: Sample-efficient learning with auxiliary class-label information. In: 2011 AMIA Annual Symposium, pp. 1004–1012. American Medical Informatics Association (2011)
28. Niaf, E., Flamary, R., Rouviere, O., Lartizien, C., Canu, S.: Kernel-based learning from both qualitative and quantitative labels: application to prostate cancer diagnosis based on multiparametric mr imaging. IEEE Trans. Image Process. **23**(3), 979–991 (2013)
29. Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362. Association for Computing Machinery (2019)
30. Pang, G., Shen, C., Jin, H., Hengel, A.V.D.: Deep weakly-supervised anomaly detection. arXiv preprint arXiv:1910.13601 (2019)
31. Peng, P., Wong, R.C.W., Yu, P.S.: Learning on probabilistic labels. In: 2014 SIAM International Conference on Data Mining, pp. 307–315. SIAM (2014)
32. Perini, L., Bürkner, P., Klami, A.: Estimating the contamination factor's distribution in unsupervised anomaly detection. In: Fortieth International Conference on Machine Learning. PMLR (2023)
33. Perini, L., Vercruyssen, V., Davis, J.: Class prior estimation in active positive and unlabeled learning. In: 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence, pp. 2915–2921. IJCAI-PRICAI (2020)
34. Perini, L., Vercruyssen, V., Davis, J.: Transferring the contamination factor between anomaly detection domains by shape similarity. In: 36th AAAI Conference on Artificial Intelligence, pp. 4128–4136. AAAI Press (2022)
35. Pimentel, T., Monteiro, M., Veloso, A., Ziviani, N.: Deep active learning for anomaly detection. In: 2020 IEEE International Joint Conference on Neural Networks, pp. 1–8. IEEE (2020)
36. Pustokhina, I., Seraj, A., Hafsan, H., Mostafavi, S.M., Alizadeh, S.: Developing a robust model based on the gaussian process regression approach to predict biodiesel properties. Int. J. Chem. Eng. 1–12 (2021)
37. Rasmussen, C.E.: Gaussian processes in machine learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) ML -2003. LNCS (LNAI), vol. 3176, pp. 63–71. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28650-9_4
38. Ratner, A., Hancock, B., Dunnmon, J., Goldman, R., Ré, C.: Snorkel metal: weak supervision for multi-task learning. In: Second Workshop on Data Management for End-to-End Machine Learning. Association for Computing Machinery (2018)
39. Raykar, V.C., et al.: Learning from crowds. J. Mach. Learn. Res. **11**(4) (2010)
40. Ruff, L., et al.: Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906.02694 (2019)
41. Russo, S., Lürig, M., Hao, W., Matthews, B., Villez, K.: Active learning for anomaly detection in environmental data. Environ. Model. Softw. **134**, 104869 (2020)
42. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
43. Thiel, C.: Classification on soft labels is robust against label noise. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008. LNCS (LNAI), vol. 5177, pp. 65–73. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85563-7_14

44. Vercruyssen, V., Meert, W., Verbruggen, G., Maes, K., Baumer, R., Davis, J.: Semi-supervised anomaly detection with an application to water analytics. In: 2018 IEEE International Conference on Data Mining, pp. 527–536. IEEE (2018)
45. Vercruyssen, V., Perini, L., Meert, W., Davis, J.: Multi-domain active learning for semi-supervised anomaly detection. In: 2022 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 485–501. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-26412-2_30
46. Xuan, J., Lu, J., Zhang, G.: A survey on Bayesian nonparametric learning. ACM Comput. Surv. **52**(1), 1–36 (2019)
47. Zhao, Y., Hryniewicki, M.K.: Xgbod: improving supervised outlier detection with unsupervised representation learning. In: 2018 IEEE International Joint Conference on Neural Networks, pp. 1–8. IEEE (2018)
48. Zhao, Z., et al.: Enhancing robustness of on-line learning models on highly noisy data. IEEE Trans. Depend. Secure Comput. **18**(05), 2177–2192 (2021)