ADBench: Anomaly Detection Benchmark

Songqiao Han[1,*], **Xiyang Hu**[2,*], **Hailiang Huang**[1,*†], **Minqi Jiang**[1,*], **Yue Zhao**[2,*†]
[1] Shanghai University of Finance and Economics [2] Carnegie Mellon University
{han.songqiao,hlhuang}@shufe.edu.cn, {2020310191}@live.sufe.edu.cn,
{xiyanghu,zhaoy}@cmu.edu

Abstract

Given a long list of anomaly detection algorithms developed in the last few decades, how do they perform with regard to (*i*) varying levels of supervision, (*ii*) different types of anomalies, and (*iii*) noisy and corrupted data? In this work, we answer these key questions by conducting (to our best knowledge) the most comprehensive anomaly detection benchmark with 30 algorithms on 57 benchmark datasets, named ADBench. Our extensive experiments (98,436 in total) identify meaningful insights into the role of supervision and anomaly types, and unlock future directions for researchers in algorithm selection and design. With ADBench, researchers can efficiently conduct comprehensive and fair evaluations for newly proposed methods on the datasets (including our contributed ones from natural language and computer vision domains) against the existing baselines. To foster accessibility and reproducibility, we fully open-source ADBench and the corresponding results.

1 Introduction

Anomaly detection (AD), which is also known as outlier detection, is a key machine learning (ML) task with numerous applications, including anti-money laundering [94], rare disease detection [196], social media analysis [186, 193], and intrusion detection [88]. AD algorithms aim to identify data instances that deviate significantly from the majority of data objects [59, 139, 146, 160], and numerous methods have been developed in the last few decades [3, 85, 102, 103, 129, 156, 172, 198]. Among them, the majority are designed for tabular data (i.e., no time dependency and graph structure). Thus, we focus on the *tabular* AD algorithms and datasets in this work.

Although there are already some benchmark and evaluation works for tabular AD [25, 38, 42, 53, 166], they generally have the limitations as follows: (*i*) primary emphasis on unsupervised methods only without including emerging (semi-)supervised AD methods; (*ii*) limited analysis of the algorithm performance concerning anomaly types (e.g., local vs. global); *(iii)* the lack of analysis on model robustness (e.g., noisy labels and irrelevant features); *(iv)* the absence of using statistical tests for algorithm comparison; and *(v)* no coverage of more complex CV and NLP datasets, which have attracted extensive attention nowadays.

To address these limitations, we design (to our best knowledge) the most comprehensive tabular anomaly detection benchmark called ADBench. By analyzing both research needs and deployment requirements in the industry, we design the experiments with three major angles in anomaly detection (see §3.3): (*i*) the availability of supervision (e.g., ground truth labels) by including 14 unsupervised, 7 semi-supervised, and 9 supervised methods; (*ii*) algorithm performance under different types of anomalies by simulating the environments with four types of anomalies; and (*iii*) algorithm robustness and stability under three settings of data corruptions. Fig. 1 provides an overview of ADBench.

Key takeaways: Through extensive experiments, we find (*i*) surprisingly none of the benchmarked unsupervised algorithms is statistically better than others, emphasizing the importance of algorithm

[*] All authors contribute equally. Names are listed in alphabetical ordering by the last name.
[†] Corresponding authors. Direct technical questions to Minqi Jiang and Yue Zhao.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

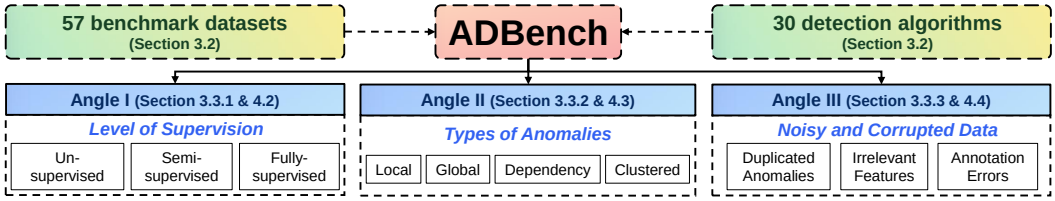


Figure 1: The design of the proposed ADBench is driven by research and application needs.

selection; (*ii*) with merely 1% labeled anomalies, most semi-supervised methods can outperform the best unsupervised method, justifying the importance of supervision; (*iii*) in controlled environments, we observe that the best unsupervised methods for specific types of anomalies are even better than semi- and fully-supervised methods, revealing the necessity of understanding data characteristics; (*iv*) semi-supervised methods show potential in achieving robustness in noisy and corrupted data, possibly due to their efficiency in using labels and feature selection. See §4 for additional results and insights.

We summarize the primary contributions of ADBench as below:

1. **The most comprehensive AD benchmark**. ADBench examines 30 detection algorithms' performance on 57 benchmark datasets (of which 47 are existing ones and we create 10).
2. **Research and application-driven benchmark angles**. By analyzing the needs of research and real-world applications, we focus on three critical comparison angles: availability of supervision, anomaly types, and algorithm robustness under noise and data corruption.
3. **Insights and future directions for researchers and practitioners**. With extensive results, we show the necessity of algorithm selection, and the value of supervision and prior knowledge.
4. **Fair and accessible AD evaluation**. We open-source ADBench with BSD-2 License at `https://github.com/Minqi824/ADBench`, for benchmarking newly proposed methods.

2 Related Work

2.1 Anomaly Detection Algorithms

Unsupervised Methods by Assuming Anomaly Data Distributions. *Unsupervised AD methods are proposed with different assumptions of data distribution* [3], e.g., anomalies located in low-density regions, and their performance depends on the agreement between the input data and the algorithm assumption(s). Many unsupervised methods have been proposed in the last few decades [3, 15, 129, 150, 198], which can be roughly categorized into shallow and deep (neural network) methods. The former often carries better interpretability, while the latter handles large, high-dimensional data better. Please see Appx. §A.1, recent book [3], and surveys [129, 150] for additional information.

Supervised Methods by Treating Anomaly Detection as Binary Classification. *With the accessibility of full ground truth labels (which is rare), supervised classifiers may identify known anomalies at the risk of missing unknown anomalies*. Arguably, there are no specialized supervised anomaly detection algorithms, and people often use existing classifiers for this purpose [3, 170] such as Random Forest [21] and neural networks [89]. One known risk of supervised methods is that ground truth labels are not necessarily sufficient to capture all types of anomalies during annotation. These methods are therefore limited to detecting unknown types of anomalies [3]. Recent machine learning books [4, 54] and scikit-learn [133] may serve as good sources of supervised ML methods.

Semi-supervised Methods with Efficient Use of Labels. *Semi-supervised AD algorithms can capitalize the supervision from partial labels, while keeping the ability to detect unseen types of anomalies.* To this end, some recent studies investigate using partially labeled data for improving detection performance and leveraging unlabeled data to facilitate representation learning. For instance, some semi-supervised models are trained only on normal samples, and detect anomalies that deviate from the normal representations learned in the training process [7, 8, 188]. In ADBench, semi-supervision mostly refers to *incomplete label learning* in weak-supervision (see [206]). More discussions on semi-supervised AD are deferred to Appx. §A.3.

2.2 Existing Datasets and Benchmarks for Tabular AD

AD Datasets in Literature. Existing benchmarks mainly evaluate a part of the datasets derived from the ODDS Library [145], DAMI Repository [25], ADRepository [129], and Anomaly Detection

Table 1: Comparison among ADBench and existing benchmarks, where ADBench comprehensively includes the most datasets and algorithms, uses both benchmark and synthetic datasets, covers both shallow and deep learning (DL) algorithms, and considers multiple comparison angles.

Benchmark	Coverage (§3.2)		Data Source		Algorithm Type		Comparison Angle (§3.3)		
	# datasets	# algo.	Real-world	Synthetic	Shallow	DL	Supervision	Types	Robustness
Ruff et al. [150]	3	9	✓	✓	✓	✓	✗	✓	✗
Goldstein et al. [53]	10	19	✓	✗	✓	✗	✗	✓	✗
Domingues et al. [38]	15	14	✓	✗	✓	✗	✗	✗	✓
Soenen et al. [164]	16	6	✓	✗	✓	✗	✗	✗	✗
Steinbuss et al. [166]	19	4	✗	✓	✓	✗	✗	✓	✗
Emmott et al. [42]	19	8	✓	✓	✓	✗	✗	✓	✓
Campos et al. [25]	23	12	✓	✗	✓	✗	✗	✗	✗
ADBench (ours)	57	30	✓	✓	✓	✓	✓	✓	✓

Meta-Analysis Benchmarks [42]. In ADBench, we include almost all publicly available datasets, and add larger datasets adapted from CV and NLP domains, for a more holistic view. See details in §3.2.

Existing Benchmarks. There are some notable works that take effort to benchmark AD methods on tabular data, e.g., [25, 38, 42, 150, 166] (see Appx. A.4). How does ADBench differ from them?

First, previous studies mainly focus on benchmarking the shallow unsupervised AD methods. Considering the rapid advancement of ensemble learning and deep learning methods, we argue that a comprehensive benchmark should also consider them. Second, most existing works only evaluate public benchmark datasets and/or some fully synthetic datasets; we organically incorporate both of them to unlock deeper insights. More importantly, existing benchmarks primarily focus on direct performance comparisons, while the settings may not be sufficiently complex to understand AD algorithm characteristics. We strive to address the above issues in ADBench, and illustrate the main differences between the proposed ADBench and existing AD benchmarks in Table 1.

Also, "anomaly detection" is an overloaded term; there are AD benchmarks for time-series [85, 87, 132], graph [101], CV [6, 27, 203] and NLP [143], but they are different from tabular AD in nature.

2.3 Connections with Related Fields and Other Opportunities

While ADBench focuses on the AD tasks, we note that there are some closely related problems, including out-of-distribution (OOD) detection [182, 183], novelty detection [116, 137], and open-set recognition (OSR) [51, 112]. Uniquely, AD usually does not assume the train set is anomaly-free, while other related tasks may do. Some methods designed for these related fields, e.g., OCSVM [157], can be used for AD as well; future benchmark can consider including: (*i*) OOD methods: MSP [65], energy-based EBO [104], and Mahalanobis distance-based MDS [92]; (*ii*) novelty detection methods: OCGAN [135] and Adversarial One-Class Classifier [154]; and (*iii*) OSR methods: OpenGAN [79] and PROSER [204]. See [155] for deeper connections and differences between AD and these fields.

We consider saliency detection (SD) [44, 46] and camouflage detection (CD) [45] as good inspirations and applications of AD tasks. Saliency detection identifies important regions in the images, where explainable AD algorithms [123], e.g., FCDD [106], may help the task. Camouflage detection finds concealed objects in the background, e.g., camouflaged anomalies blurred with normal objects [110], where camouflage-resistant AD methods [40] help detect concealed objects (that look normal but are abnormal). Future work can explore the explainability of detected objects in AD.

3 ADBench: AD Benchmark Driven by Research and Application Needs

3.1 Preliminaries and Problem Definition

Unsupervised AD often presents a collection of n samples $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^{n \times d}$, where each sample has d features. Given the inductive setting, the goal is to train an AD model M to output anomaly score $\mathbf{O} \coloneqq M(\mathbf{X}) \in \mathbb{R}^{n \times 1}$, where higher scores denote for more outlyingness. In the inductive setting, we need to predict on $\mathbf{X}_{\text{test}} \in \mathbb{R}^{m \times d}$, so to return $\mathbf{O}_{\text{test}} \coloneqq M(\mathbf{X}_{\text{test}}) \in \mathbb{R}^{m \times 1}$.

Supervised AD also has the (binary) ground truth labels of $\mathbf{X}$, i.e., $\mathbf{y} \in \mathbb{R}^{n \times 1}$. A supervised AD model M is first trained on $\{\mathbf{X}, \mathbf{y}\}$, and then returns anomaly scores for the $\mathbf{O}_{\text{test}} \coloneqq M(\mathbf{X}_{\text{test}})$.

Semi-supervised AD only has the partial label information $\mathbf{y}^l \in \mathbf{y}$. The AD model M is trained on the entire feature space $\mathbf{X}$ with the partial label $\mathbf{y}^l$, i.e., $\{\mathbf{X}, \mathbf{y}^l\}$, and then outputs $\mathbf{O}_{\text{test}} \coloneqq M(\mathbf{X}_{\text{test}})$.

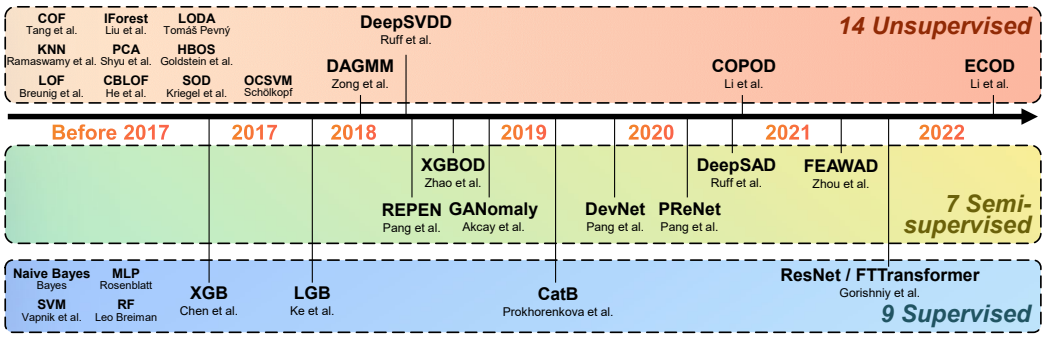


Figure 2: ADBench covers a wide range of AD algorithms. See Appx. B.1 for more details.

Remark. Irrespective of the types of underlying AD algorithms, the goal of ADBench is to understand AD algorithms' performance under the inductive setting. Collectively, we refer semi-supervised and supervised AD methods as "label-informed" methods. Refer to §4.1 for specific experiment settings.

3.2 The Largest AD Benchmark with 30 Algorithms and 57 Datasets

Algorithms. Compared to the previous benchmarks, we have a larger algorithm collection with (*i*) the latest unsupervised AD algorithms like DeepSVDD [151] and ECOD [97]; (*ii*) SOTA semi-supervised algorithms, including DeepSAD [152] and DevNet [131]; (*iii*) latest network architectures like ResNet [62] in computer vision (CV) and Transformer [171] in the natural language processing (NLP) domain—we adapt ResNet and FTTransformer models [56] for tabular AD in the proposed ADBench; and (*iv*) ensemble learning methods like LightGBM [74], XGBoost [29], and CatBoost [138] that have shown effectiveness in AD tasks [170]. Fig. 2 shows the 30 algorithms (14 unsupervised, 7 semi-supervised, and 9 supervised algorithms) evaluated in ADBench, where we provide more information about them in Appx. B.1.

Algorithm Implementation. Most unsupervised algorithms are readily available in our early work Python Outlier Detection (PyOD) [198], and some supervised methods are available in scikit-learn [133] and corresponding libraries. Supervised ResNet and FTTransformer tailored for tabular data have been open-sourced in their original paper [56]. We implement the semi-supervised methods and release them along with ADBench.

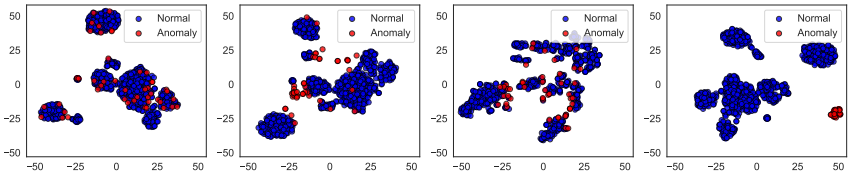
Public AD Datasets. In ADBench, we gather more than 40 benchmark datasets [25, 42, 129, 145], for model evaluation, as shown in Appx. Table B1. These datasets cover many application domains, including healthcare (e.g., disease diagnosis), audio and language processing (e.g., speech recognition), image processing (e.g., object identification), finance (e.g., financial fraud detection), etc. For due diligence, we keep the datasets where the anomaly ratio is below 40% (Appx. Fig. B1).

Newly-added Datasets in ADBench. Since most of these datasets are relatively small, we introduce 10 more complex datasets from CV and NLP domains with more samples and richer features in ADBench (highlighted in Appx. Table B1). Pretrained models are applied to extract data embedding from CV and NLP datasets to access more complex representations, which has been widely used in AD literature [33, 115, 152] and shown better results than using the raw features. For NLP datasets, we use BERT [75] pretrained on the BookCorpus and English Wikipedia to extract the embedding of the [CLS] token. For CV datasets, we use ResNet18 [62] pretrained on the ImageNet [35] to extract the embedding after the last average pooling layer. Following previous works [151, 152], we set one of the multi-classes as normal, downsample the remaining classes to 5% of the total instances as anomalies, and report the average results over all the respective classes. Including these originally non-tabular datasets helps to see whether tabular AD methods can work on CV/NLP data after necessary preprocessing. See Appx. B.2 for more details on datasets.

3.3 Benchmark Angles in ADBench

3.3.1 Angle I: Availability of Ground Truth Labels (Supervision)

Motivation. As shown in Table 1, existing benchmarks only focus on the unsupervised setting, i.e., none of the labeled anomalies is available. Despite, in addition to unlabeled samples, one may have access to a limited number of labeled anomalies in real-world applications, e.g., a few anomalies identified by domain experts or human-in-the-loop techniques like active learning [5, 7, 78, 189].



(a) Local (b) Global (c) Dependency (d) Clustered

Figure 3: Illustration of four types of synthetic anomalies shown on Lymphography dataset. See the additional demo in Appx. Fig. B2.

Notably, there is a group of semi-supervised AD algorithms [127, 128, 130, 131, 152, 168, 205] that have not been covered by existing benchmarks.

Our design: We first benchmark existing unsupervised anomaly detection methods, and then evaluate both semi-supervised and fully-supervised methods with varying levels of supervision following the settings in [127, 131, 205] to provide a fair comparison. For example, labeled anomalies $\gamma_l = 10\%$ means that 10% anomalies in the train set are known while other samples remain unlabeled. The complete experiment results of un-, semi-, and full-supervised algorithms are presented in §4.2.

3.3.2 Angle II: Types of Anomalies

Motivation. While extensive public datasets can be used for benchmarking, they often consist of a mixture of different types of anomalies, making it challenging to understand the pros and cons of AD algorithms regarding specific types of anomalies [55, 166]. In real-world applications, one may know specific types of anomalies of interest. To better understand the impact of anomaly types, we create synthetic datasets based on public datasets by injecting specific types of anomalies to analyze the response of AD algorithms.

Our design: In ADBench, we create *realistic* synthetic datasets from benchmark datasets by injecting specific types of anomalies. Some existing works, such as PyOD [198], generate fully synthetic anomalies by assuming their data distribution, which fails to create complex anomalies. We follow and enrich the approach in [166] to generate "realistic" synthetic data; ours supports more types of anomaly generation. The core idea is to build a generative model (e.g., Gaussian mixture model GMM used in [166], Sparx [191], and ADBench) using the normal samples from a benchmark dataset and discard its original anomalies as we do not know their types. Then, We could generate normal samples and different types of anomalies based on their definitions by tweaking the generative model. The generation of normal samples is the same in all settings if not noted, and we provide the generation process of four types of anomalies below (also see our codebase for details).

Definition and Generation Process of Four Types of Common Anomalies Used in ADBench:

- **Local anomalies** refer to the anomalies that are deviant from their local neighborhoods [22]. We follow the GMM procedure [118, 166] to generate synthetic normal samples, and then scale the covariance matrix $\hat{\mathbf{\Sigma}} = \alpha\hat{\mathbf{\Sigma}}$ by a scaling parameter $\alpha = 5$ to generate local anomalies.
- **Global anomalies** are more different from the normal data [68], generated from a uniform distribution Unif $\left(\alpha \cdot \min\left(\mathbf{X^k}\right), \alpha \cdot \max\left(\mathbf{X^k}\right)\right)$, where the boundaries are defined as the *min* and *max* of an input feature, e.g., k-th feature $\mathbf{X^k}$, and $\alpha = 1.1$ controls the outlyingness of anomalies.
- **Dependency anomalies** refer to the samples that do not follow the dependency structure which normal data follow [117], i.e., the input features of dependency anomalies are assumed to be independent of each other. Vine Copula [1] method is applied to model the dependency structure of original data, where the probability density function of generated anomalies is set to complete independence by removing the modeled dependency (see [117]). We use Kernel Density Estimation (KDE) [61] to estimate the probability density function of features and generate normal samples.
- **Clustered anomalies**, also known as group anomalies [93], exhibit similar characteristics [42, 99]. We scale the mean feature vector of normal samples by $\alpha = 5$, i.e., $\hat{\boldsymbol{\mu}} = \alpha\hat{\boldsymbol{\mu}}$, where α controls the distance between anomaly clusters and the normal, and use the scaled GMM to generate anomalies.

Fig. 3 shows 2-d t-SNE [169] visualization of the four types of synthetic outliers generated from Lymphography dataset, where they generally satisfy the expected characteristics. Local anomalies (Fig. 3a) are well overlapped with the normal samples. Global anomalies (Fig. 3b) are more deviated from the normal samples and on the edges of normal clusters. The other two types of anomalies are as expected, with no clear dependency structure in Fig. 3c and having anomaly cluster(s) in Fig. 3d. In ADBench, we analyze the algorithm performances under all four types of anomalies above (§4.3).

3.3.3 Angle III: Model Robustness with Noisy and Corrupted Data

Motivation. Model robustness has been an important aspect of anomaly detection and adversarial machine learning [24, 41, 47, 76, 177]. Meanwhile, the input data likely suffers from noise and corruption to some extent in real-world applications [42, 55, 60, 124]. However, this important view has not been well studied in existing benchmarks, and we try to understand this by evaluating AD algorithms under three noisy and corruption settings (see results in §4.4):

- **Duplicated Anomalies**. In many applications, certain anomalies likely repeat multiple times in the data for reasons such as recording errors [83]. The presence of duplicated anomalies is also called the "anomaly masking" [55, 60, 100], posing challenges to many AD algorithms [25], e.g., the density-based KNN [11, 144]. Besides, the change of anomaly frequency would also affect the behavior of detection methods [42]. Therefore, we simulate this setting by splitting the data into train and test set, then duplicating the anomalies (both features and labels) up to 6 times in both sets, and observing how AD algorithms change.
- **Irrelevant Features**. Tabular data may contain irrelevant features caused by measurement noise or inconsistent measuring units [28, 55], where these noisy dimensions could hide the characteristics of anomaly data and thus make the detection process more difficult [128, 150]. We add irrelevant features up to 50% of the total input features (i.e., d in the problem definition) by generating uniform noise features from Unif $\left(\min\left(\mathbf{X^k}\right), \max\left(\mathbf{X^k}\right)\right)$ of randomly selected k-th input feature $\mathbf{X^k}$ while the labels stay correct, and summarize the algorithm performance changes.
- **Annotation Errors**. While existing studies [131, 152] explored anomaly contamination in the unlabeled samples, we further discuss the more generalized impact of label contamination on the algorithm performance, where the label flips [122, 202] between the normal samples and anomalies are considered (up to 50% of total labels). Note this setting does not affect unsupervised methods as they do not use any labels. Discussion of annotation errors is meaningful since manual annotation or some automatic labeling techniques are always noisy while being treated as perfect.

4 Experiment Results and Analyses

We conduct 98,436 experiments (Appx. C) to answer **Q1** (§4.2): How do AD algorithms perform with varying levels of supervision? **Q2** (§4.3): How do AD algorithms respond to different types of anomalies? **Q3** (§4.4): How robust are AD algorithms with noisy and corrupted data? In each subsection, we first present the key results and analyses (please refer to the additional points in Appx. D), and then propose a few open questions and future research directions.

4.1 Experiment Setting

Datasets, Train/test Data Split, and Independent Trials. As described in §3.2 and Appx. Table B1, ADBench includes 57 existing and freshly proposed datasets, which cover different fields including healthcare, security, and more. Although unsupervised AD algorithms are primarily designed for the transductive setting (i.e., outputting the anomaly scores on the input data only other than making predictions on the newcoming data), we adapt all the algorithms for the inductive setting to predict the newcoming data, which is helpful in applications and also common in popular AD library PyOD [198], TODS [84], and PyGOD [102]. Thus, we use 70% data for training and the remaining 30% as the test set. We use stratified sampling to keep the anomaly ratio consistent. We repeat each experiment 3 times and report the average. Detailed settings are described in Appx. C.

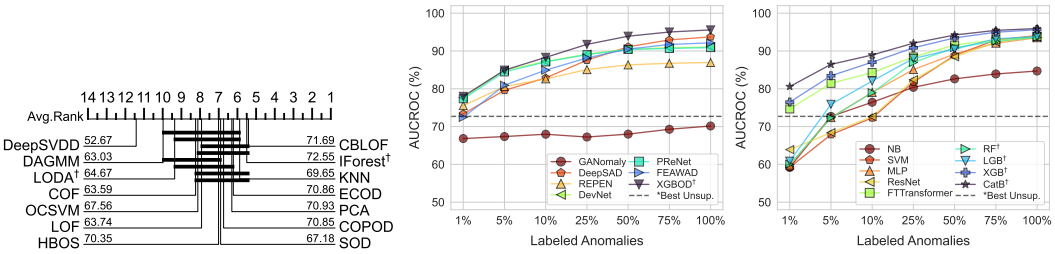
Hyperparameter Settings. For all the algorithms in ADBench, we use their default hyperparameter (HP) settings in the original paper for a fair comparison. Refer to the Appx. C for more information.

Evaluation Metrics and Statistical Tests. We evaluate different AD methods by two widely used metrics: AUCROC (Area Under Receiver Operating Characteristic Curve) and AUCPR (Area Under Precision-Recall Curve) value[1]. Besides, the critical difference diagram (CD diagram) [34, 70] based on the Wilcoxon-Holm method is used for comparing groups of AD methods statistically ($p \leq 0.05$).

4.2 Overall Model Performance on Datasets with Varying Degrees of Supervision

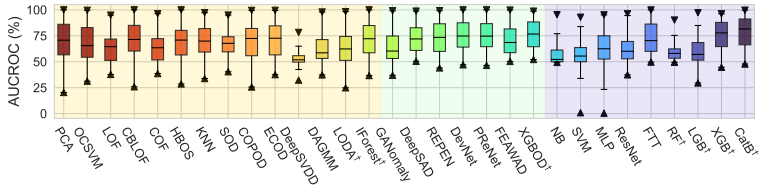
As introduced in §3.3.1, we first present the results of unsupervised methods on 57 datasets in Fig. 4a, and then compare label-informed semi- and fully-supervised methods under varying degrees of supervision, i.e., different label ratios of γ_l (from 1% to 100% full labeled anomalies) in Fig. 4b.

[1]We present the results based on AUCROC and observe similar results for AUCPR; See Appx. D for all.

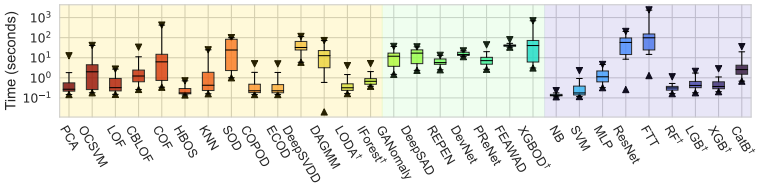


(a) Avg. rank (lower the better) and avg. AUCROC (on each line) of unsupervised methods; groups of algorithms not statistically different are connected horizontally.

(b) Avg. AUCROC (on 57 datasets) vs. % of labeled anomalies (x-axis); semi-supervised (left) and fully-supervised (right). Most label-informed algorithms outperform the best *unsupervised* algorithm CBLOF (denoted as the dashed line) with 10% labeled anomalies.



(c) Boxplot of AUCROC (@1% labeled anomalies) on 57 datasets; we denote un-, semi-, and fully supervised methods in light yellow, green, and purple.



(d) Boxplot of train time (see inf. time in Appx. Fig. D6) on 57 datasets; we denote un-, semi-, and fully supervised methods in light yellow, green, and purple.

Figure 4: Average AD model performance across 57 benchmark datasets. (a) shows that no unsupervised algorithm statistically outperforms the rest. (b) shows that semi-supervised methods leverage the labels more efficiently than fully-supervised methods with a small labeled anomaly ratio γ_l. (c) and (d) present the boxplots of AUCROC and runtime. Ensemble methods are marked with "†".

None of the unsupervised methods is statistically better than the others, as shown in the critical difference diagram of Fig. 4a (where most algorithms are horizontally connected without statistical significance). We also note that some DL-based unsupervised methods like DeepSVDD and DAGMM are surprisingly worse than shallow methods. Without the guidance of label information, DL-based unsupervised algorithms are harder to train (due to more hyperparameters) and more difficult to tune hyperparameters, leading to unsatisfactory performance.

Semi-supervised methods outperform supervised methods when limited label information is available. For $\gamma_l \leq 5\%$, i.e., only less than 5% labeled anomalies are available during training, the detection performance of semi-supervised methods (median AUCROC$= 75.56\%$ for $\gamma_l = 1\%$ and AUCROC$= 80.95\%$ for $\gamma_l = 5\%$) are generally better than that of fully-supervised algorithms (median AUCROC$= 60.84\%$ for $\gamma_l = 1\%$ and AUCROC$= 72.69\%$ for $\gamma_l = 5\%$). For most semi-supervised methods, merely 1% labeled anomalies are sufficient to surpass the best unsupervised method (shown as the dashed line in Fig. 4b), while most supervised methods need 10% labeled anomalies to achieve so. We also show the improvement of algorithm performances about the increasing γ_l, and notice that with a large number of labeled anomalies, both semi-supervised and supervised methods have comparable performance. Putting these together, we verify the assumed advantage of semi-supervised methods in leveraging limited label information more efficiently.

Latest network architectures like Transformer and emerging ensemble methods yield competitive performance in AD. Fig. 4b shows FTTransformer and ensemble methods like XGB(oost) and CatB(oost) provide satisfying detection performance among all the label-informed algorithms, even these methods are not specifically proposed for the anomaly detection tasks. For $\gamma_l = 1\%$, the AUCROC of FTTransformer and the median AUCROC of ensemble methods are 74.68% and 76.47%, respectively, outperforming the median AUCROC of all label-informed methods 72.91%. The great performance of tree-based ensembles (in tabular AD) is consistent with the findings in literature [20, 58, 170], which may be credited to their capacity to handle imbalanced AD datasets via aggregation. Future research may focus on understanding the cause and other merits of ensemble trees in tabular AD, e.g., better model efficiency.

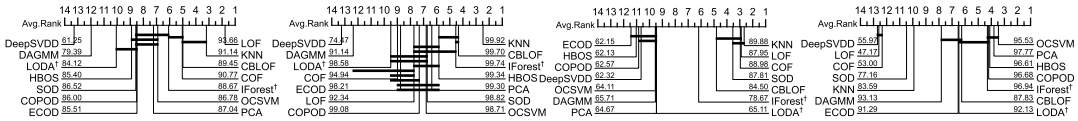
Runtime Analysis. We present the train and inference time in Fig. 4d and Appx. Fig. D6. Runtime analysis finds that HBOS, COPOD, ECOD, and NB are the fastest as they treat each feature independently. In contrast, more complex representation learning methods like XGBOD, ResNet, and FTTansformer are computationally heavy. This should be factored in for algorithm selection.

Future Direction 1: Unsupervised Algorithm Evaluation, Selection, and Design. For unsupervised AD, the results suggest that future algorithms should be evaluated on large testbeds like ADBench for statistical tests (such as via critical difference diagrams). Meanwhile, the no-free-lunch theorem [175] suggests there is no universal winner for all tasks, and more focus should be spent on understanding the suitability of each AD algorithm. Notably, algorithm selection and hyperparameter optimization are important in unsupervised AD, but limited works [13, 109, 194, 199, 200] have studied them. We may consider self-supervision [140, 158, 161, 179] and transfer learning [33] to improve tabular AD as well. Thus, we call for attention to large-scale evaluation, task-driven algorithm selection, and data augmentation/transfer for unsupervised AD.

Future Direction 2: Semi-supervised Learning. By observing the success of using limited labels in AD, we would call for more attention to semi-supervised AD methods which can leverage both the guidance from labels efficiently and the exploration of the unlabeled data. Regarding backbones, the latest network architectures like Transformer and ensembling show their superiority in AD tasks.

4.3 Algorithm Performance under Different Types of Anomalies

Under four types of anomalies introduced in §3.3.2), we show the performances of unsupervised methods in Fig. 5, and then compare both semi- and fully-supervised methods in Fig. 6.



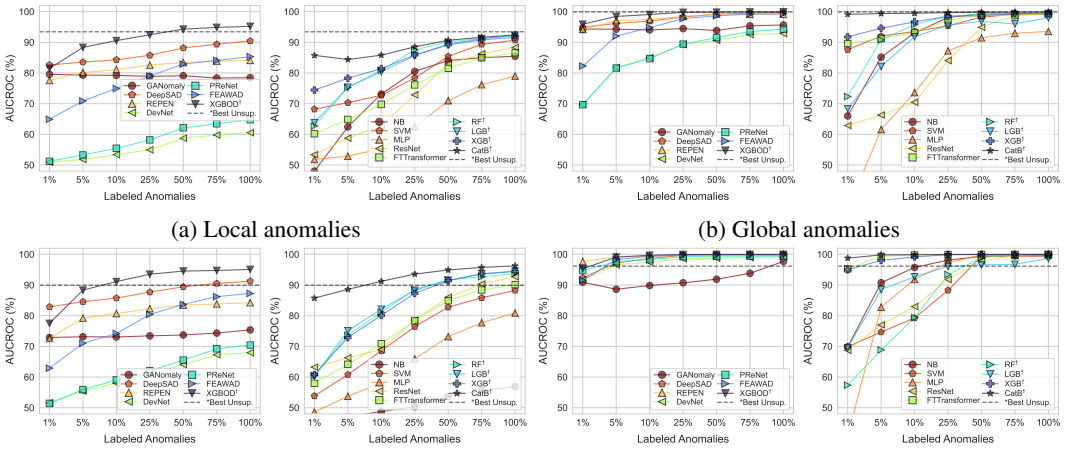
(a) Local anomalies (b) Global anomalies (c) Dependency anomalies (d) Clustered anomalies

Figure 5: Avg. rank (lower the better) of unsupervised methods on different types of anomalies. Groups of algorithms not significantly different are connected horizontally in the CD diagrams. The unsupervised methods perform well when their assumptions conform to the underlying anomaly type.

Performance of unsupervised algorithms highly depends on the alignment of its assumptions and the underlying anomaly type. As expected, *local* anomaly factor (LOF) is statistically better than other unsupervised methods for the local anomalies (Fig. 5a), and KNN, which uses k-th (*global*) nearest neighbor's distance as anomaly scores, is the statistically best detector for global anomalies (Fig. 5b). Again, there is no algorithm performing well on all types of anomalies; LOF achieves the best AUCROC on local anomalies (Fig. 5a) and the second best AUCROC rank on dependency anomalies (Fig. 5c), but performs poorly on clustered anomalies (Fig. 5d). Practitioners should select algorithms based on the characteristics of the underlying task, and consider the algorithm which may cover more high-interest anomaly types [93].

The "power" of prior knowledge on anomaly types may overweigh the usage of partial labels. For the local, global, and dependency anomalies, most label-informed methods perform worse than the best unsupervised methods of each type (corresponding to LOF, KNN, and KNN). For example, the detection performance of XGBOD for the local anomalies is inferior to the best unsupervised method LOF when $\gamma_l \leq 50\%$, while other methods perform worse than LOF in all cases (See Fig. 6a). Why could not label-informed algorithms beat unsupervised methods in this setting? We believe that partially labeled anomalies cannot well capture all characteristics of specific types of anomalies, and learning such decision boundaries is challenging. For instance, different local anomalies often exhibit various behaviors, as shown in Fig. 3a, which may be easier to identify by a generic definition of "locality" in unsupervised methods other than specific labels. Thus, incomplete label information may bias the learning process of these label-informed methods, which explains their relatively inferior performances compared to the best unsupervised methods. This conclusion is further verified by the results of clustered anomalies (See Fig. 6d), where label-informed (especially semi-supervised) methods outperform the best unsupervised method OCSVM, as few labeled anomalies can already represent similar behaviors in the clustered anomalies (Fig. 3d).

Future Direction 3: Leveraging Anomaly Types as Valuable Prior Knowledge. The above results emphasize the importance of knowing anomaly types in achieving high detection performance even without labels, and call for attention to designing anomaly-type-aware detection algorithms. In an



(a) Local anomalies (b) Global anomalies

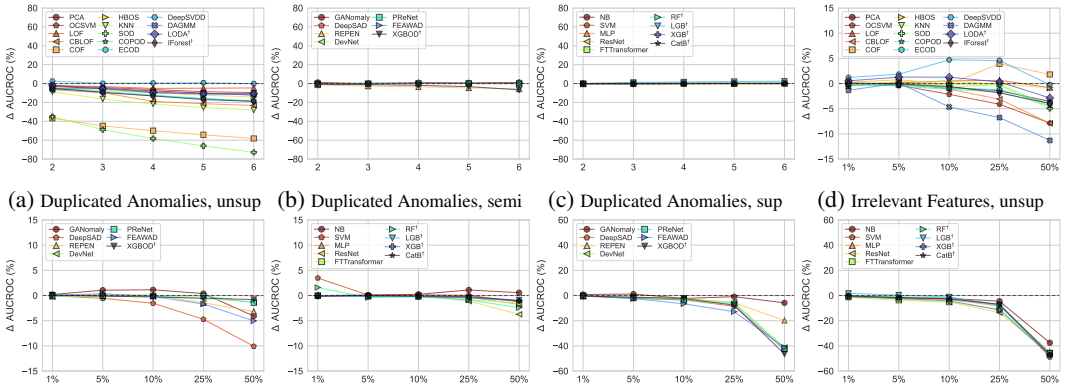
(c) Dependency anomalies (d) Clustered anomalies

Figure 6: Semi- (left of each subfigure) and supervised (right) algorithms' performance on different types of anomalies with varying levels of labeled anomalies. Surprisingly, these label-informed algorithms are *inferior* to the best unsupervised method except for the clustered anomalies.

ideal world, one may combine multiple AD algorithms based on the composition of anomaly types, via frameworks like dynamic model selection and combination [197]. To our knowledge, the latest advancement in this end [71] provides an equivalence criterion for measuring to what degree two anomaly detection algorithms detect the same kind of anomalies. Furthermore, future research may also consider designing semi-supervised AD methods capable of detecting different types of unknown anomalies while effectively improving performance by the partially available labeled data. Another interesting direction is to train an offline AD model using synthetically generated anomalies and then adapt it for online prediction on real-world datasets with likely similar anomaly types. Unsupervised domain adaption and transfer learning for AD [33, 185] may serve as useful references.

4.4 Algorithm Robustness under Noisy and Corrupted Data

In this section, we investigate the algorithm robustness (i.e., Δperformance; see absolute performance plot in Appx. D9) of different AD algorithms under noisy and data corruption described in §3.3.3. The default γ_l is set to 100% since we only care about the relative change of model performance. Fig. 7 demonstrates the results.



(a) Duplicated Anomalies, unsup (b) Duplicated Anomalies, semi (c) Duplicated Anomalies, sup (d) Irrelevant Features, unsup

(e) Irrelevant Features, semi (f) Irrelevant Features, sup (g) Annotation Errors, semi (h) Annotation Errors, sup

Figure 7: Algorithm performance change under noisy and corrupted data (i.e., duplicated anomalies for (a)-(c), irrelevant features for (d)-(f), and annotation errors for (g) and (h)). X-axis denotes either the duplicated times or the noise ratio. Y-axis denotes the % of performance change (ΔAUCROC), and its range remains consistent across different algorithms. The results reveal unsupervised methods' susceptibility to duplicated anomalies and the usage of label information in defending irrelevant features. Un-, semi-, and fully-supervised methods are denoted as *unsup*, *semi*, and *sup*, respectively.

Unsupervised methods are more susceptible to duplicated anomalies. As shown in Fig. 7a, almost all unsupervised methods are severely impacted by duplicated anomalies. Their AUCROC

deteriorates proportionally with the increase in duplication. When anomalies are duplicated by 6 times, the median ΔAUCROC of unsupervised methods is -16.43%, compared to that of semi-supervised methods -0.05% (Fig. 7b) and supervised methods 0.13% (Fig. 7c). One explanation is that unsupervised methods often assume the underlying data is imbalanced with only a smaller percentage of anomalies—they rely on this assumption to detect anomalies. With more duplicated anomalies, the underlying data becomes more balanced, and the minority assumption of anomalies is violated, causing the degradation of unsupervised methods. Differently, more balanced datasets do not affect the performance of semi- and fully-supervised methods remarkably, with the help of labels.

Irrelevant features cause little impact on supervised methods due to feature selection. Compared to the unsupervised and most semi-supervised methods, the training process of supervised methods is fully guided by the data labels ($\mathbf{y}$), therefore performing robustly to the irrelevant features (i.e., corrupted $\mathbf{X}$) due to the direct (or indirect) feature selection process. For instance, ensemble trees like XGBoost can filter irrelevant features. As shown in Fig. 7f, even the worst performing supervised algorithm (say ResNet) in this setting yields $\leq 5\%$ degradation when 50% of the input features are corrupted by the uniform noises, while the un- and semi-supervised methods could face up to 10% degradation. Besides, the robust performances of supervised methods (and some semi-supervised methods like DevNet) indicate that the label information can be beneficial for feature selection. Also, Fig. 7f shows that minor irrelevant features (e.g., 1%) help supervised methods as regularization to generalize better.

Both semi- and fully-supervised methods show great resilience to minor annotation errors. Although the detection performance of these methods is significantly downgraded when the annotation errors are severe (as shown in Fig. 7g and 7h), their degradation with regard to minor annotation errors is acceptable. The median ΔAUCROC of semi- and fully-supervised methods for 5% annotation errors is -1.52% and -1.91%, respectively. That being said, label-informed methods are still acceptable in practice as the annotation error should be relatively small [95, 181].

Future Direction 4: Noise-resilient AD Algorithms. Our results indicate there is an improvement space for robust unsupervised AD algorithms. One immediate remedy is to incorporate unsupervised feature selection [30, 125, 126] to combat irrelevant features. Moreover, label information could serve as effective guidance for model training against data noise, and it helps semi- and fully-supervised methods to be more robust. Given the difficulty of acquiring full labels, we suggest using semi-supervised methods as the backbone for designing more robust AD algorithms. Also, recent works on leveraging multiple sets of noisy labels collectively for learning AD models are also relevant [201].

5 Conclusions and Future Work

In this paper, we introduce ADBench, the most comprehensive tabular anomaly detection benchmark with 30 algorithms and 57 benchmark datasets. Based on the analyses of multiple comparison angles, we unlock insights into the role of supervision, the importance of prior knowledge of anomaly types, and the principles of designing robust detection algorithms. On top of them, we summarize a few promising future research directions for anomaly detection, along with the fully released benchmark suite for evaluating new algorithms.

ADBench can extend to understand the algorithm performance with (*i*) mixed types of anomalies; (*ii*) different levels of (intrinsic) anomaly ratio; and (*iii*) more data modalities. Also, future benchmarks can consider the latest algorithms [28, 99, 161], and curate datasets from emerging fields like drug discovery [69], molecule optimization [49, 50], interpretability and explainability [123, 180], and bias and fairness [32, 67, 123, 159, 165, 190].

Aknowledgement

We briefly describe the authors' contributions. *Problem scoping*: M.J., Y.Z., S.H., X.H., and H.H.; *Experiment and Implementation*: M.J. and Y.Z.; *Result Analysis*: M.J., Y.Z., and X.H.; *Paper Drafting*: M.J., Y.Z., S.H., X.H., and H.H.; *Paper Revision*: M.J., Y.Z., S.H., and X.H.

We thank anonymous reviewers for their insightful feedback and comments. We appreciate the suggestions of Xueying Ding, Kwei-Herng (Henry) Lai, Meng-Chieh Lee, Ninghao Liu, Yuwen Yang, Allen Zhu, Chaochuan Hou, and Xu Yao. Y.Z. is partly supported by the Norton Graduate Fellowship. H.H., S.H., and M.J. are supported by the National Natural Science Foundation of China under Grant No. 72271151, 92146004, and the National Key Research and Development Program of China under Grant No. 2022YFC3303301. H.H., S.H., and M.J. thank the financial support provided by FlagInfo-SHUFE Joint Laboratory.

References

[1] K. Aas, C. Czado, A. Frigessi, and H. Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

[2] S. Aeberhard, D. Coomans, and O. de Vel. The classification performance of rda. *Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep*, pages 92–01, 1992.

[3] C. C. Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pages 1–34. 2017.

[4] C. C. Aggarwal. *Neural Networks and Deep Learning - A Textbook*. Springer, 2018.

[5] N. B. Aissa and M. Guerroumi. Semi-supervised statistical approach for network anomaly detection. *Procedia Computer Science*, 83:1090–1095, 2016.

[6] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc. Anomalib: A deep learning library for anomaly detection. *arXiv:2202.08341*, 2022.

[7] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, pages 622–637, 2018.

[8] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *IJCNN*, pages 1–8. IEEE, 2019.

[9] F. Alimoglu and E. Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In *TAINN*. Citeseer, 1996.

[10] E. Alpaydin and C. Kaynak. Cascading classifiers. *Kybernetika*, 34(4):369–374, 1998.

[11] F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *ECML/PKDD*, pages 15–27. Springer, 2002.

[12] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de Sa, and L. Pereira-Leite. Sisporto 2.0: a program for automated analysis of cardiotocograms. *Journal of Maternal-Fetal Medicine*, 2000.

[13] M. Bahri, F. Salutari, A. Putina, and M. Sozio. Automl: state of the art with a focus on anomaly detection, challenges, and research directions. *IJDSA*, pages 1–14, 2022.

[14] T. Bayes. Lii. an essay towards solving a problem in the doctrine of chances. *Philos. Trans. Royal Soc. A*, pages 370–418, 1763.

[15] L. Bergman and Y. Hoshen. Classification-based anomaly detection for general data. In *ICLR*, 2019.

[16] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, pages 9592–9600, 2019.

[17] E. Berthonnaud, J. Dimnet, P. Roussouly, and H. Labelle. Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. *Clinical Spine Surgery*, 18(1):40–47, 2005.

[18] J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017.

[20] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci. Deep neural networks and tabular data: A survey. *arXiv:2110.01889*, 2021.

[21] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[22] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *SIGMOD*, pages 93–104, 2000.

[23] N. Brümmer, S. Cumani, O. Glembek, M. Karafiát, P. Matějka, J. Pešán, O. Plchot, M. Soufifar, E. d. Villiers, and J. H. Černockỳ. Description and analysis of the brno276 system for lre2011. In *Odyssey 2012-the speaker and language recognition workshop*, 2012.

[24] H. Cai, J. Liu, and W. Yin. Learned robust pca: A scalable deep unfolding approach for high-dimensional outlier detection. *NeurIPS*, 34, 2021.

[25] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 30(4):891–927, 2016.

[26] B. Cestnik, I. Kononenko, and I. Bratko. A knowledge-elicitation tool for sophisticated users. In *European Conference on European Working Session on Learning EWSL*, volume 87, 1987.

[27] R. Chan, K. Lis, S. Uhlemeyer, H. Blum, S. Honari, R. Siegwart, P. Fua, M. Salzmann, and M. Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. In *NeurIPS*, 2021.

[28] C.-H. Chang, J. Yoon, S. Arik, M. Udell, and T. Pfister. Data-efficient and interpretable tabular anomaly detection. *ArXiv*, 2203.02034, 2022.

[29] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, pages 785–794, 2016.

[30] L. Cheng, Y. Wang, X. Liu, and B. Li. Outlier detection ensemble with embedded feature selection. In *AAAI*, volume 34, pages 3503–3512, 2020.

[31] C. Cortes and V. Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[32] I. Davidson and S. S. Ravi. A framework for determining the fairness of outlier detection. In *ECAI 2020*, pages 2465–2472. IOS Press, 2020.

[33] L. Deecke, L. Ruff, R. A. Vandermeulen, and H. Bilen. Transfer-based semantic anomaly detection. In *ICML*, pages 2546–2558, 2021.

[34] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The JMLR*, 7:1–30, 2006.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[36] P. Diaconis and B. Efron. Computer-intensive methods in statistics. *Scientific American*, 248(5), 1983.

[37] T. Dietterich, A. Jain, R. Lathrop, and T. Lozano-Perez. A comparison of dynamic reposing and tangent distance for drug activity prediction. *NeurIPS*, 6, 1993.

[38] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, 74:406–421, 2018.

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[40] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *CIKM*, pages 315–324, 2020.

[41] X. Du, J. Zhang, B. Han, T. Liu, Y. Rong, G. Niu, J. Huang, and M. Sugiyama. Learning diverse-structured networks for adversarial robustness. In *ICML*, pages 2880–2891, 2021.

[42] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong. A meta-analysis of the anomaly detection problem. *ArXiv*, 1503.01158, 2015.

[43] I. W. Evett and E. J. Spiehler. Rule induction in forensic science. pages 152–160, 1989.

[44] D.-P. Fan, M.-M. Cheng, J.-J. Liu, S.-H. Gao, Q. Hou, and A. Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*, pages 186–202, 2018.

[45] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao. Camouflaged object detection. In *CVPR*, pages 2777–2787, 2020.

[46] D.-P. Fan, T. Li, Z. Lin, G.-P. Ji, D. Zhang, M.-M. Cheng, H. Fu, and J. Shen. Re-thinking co-salient object detection. *TPAMI*, 2021.

[47] L. Fan, S. Liu, P.-Y. Chen, G. Zhang, and C. Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *NeurIPS*, 34, 2021.

[48] P. W. Frey and D. J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine learning*, 6(2):161–182, 1991.

[49] T. Fu, C. Xiao, X. Li, L. M. Glass, and J. Sun. Mimosa: Multi-constraint molecule sampling for molecule optimization. In *AAAI*, volume 35, pages 125–133, 2021.

[50] T. Fu, C. Xiao, and J. Sun. Core: Automatic molecule optimization using copy & refine strategy. In *AAAI*, volume 34, pages 638–645, 2020.

[51] C. Geng, S.-j. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2020.

[52] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 9, 2012.

[53] M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[54] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[55] P. Gopalan, V. Sharan, and U. Wieder. Pidforest: anomaly detection via partial identification. *NeurIPS*, 32, 2019.

[56] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. *NeurIPS*, 34, 2021.

[57] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Toward supervised anomaly detection. *JAIR*, 46:235–262, 2013.

[58] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv:2207.08815*, 2022.

[59] C. Grunau and V. Rozhoň. Adapting k-means algorithms for outliers. *ArXiv*, 2007.01118, 2020.

[60] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust random cut forest based anomaly detection on streams. In *ICML*, pages 2712–2721, 2016.

[61] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. 2009.

[62] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[63] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517, 2016.

[64] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003.

[65] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

[66] P. Horton and K. Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. In *Ismb*, volume 4, pages 109–115, 1996.

[67] X. Hu, Y. Huang, B. Li, and T. Lu. Uncovering the source of machine bias. *KDD 2021, Machine Learning for Consumers and Markets Workshop*, 2021.

[68] H. Huang, H. Qin, S. Yoo, and D. Yu. Physics-based anomaly detection defined on manifold space. *TKDD*, 9(2):1–39, 2014.

[69] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. Coley, C. Xiao, J. Sun, and M. Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *NeurIPS*, 2021.

[70] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[71] C. I. Jerez, J. Zhang, and M. R. Silva. On equivalence of anomaly detection algorithms. *TKDD*, 2022.

[72] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *EACL*, pages 427–431, 2017.

[73] Y. Kawachi, Y. Koizumi, and N. Harada. Complementary set variational autoencoder for supervised anomaly detection. In *ICASSP*, pages 2366–2370. IEEE, 2018.

[74] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *NeurIPS*, 30:3146–3154, 2017.

[75] J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[76] M. Kim, J. Tack, and S. J. Hwang. Adversarial self-supervised contrastive learning. *NeurIPS*, 33:2983–2994, 2020.

[77] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv*, 1412.6980, 2014.

[78] B. R. Kiran, D. M. Thomas, and R. Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

[79] S. Kong and D. Ramanan. Opengan: Open-set recognition via open data generation. In *CVPR*, pages 813–822, 2021.

[80] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *PAKDD*, pages 831–838. Springer, 2009.

[81] A. Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[82] M. Kudo, J. Toyama, and M. Shimbo. Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters*, 20(11-13):1103–1111, 1999.

[83] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim. An empirical study on network anomaly detection using convolutional neural networks. In *ICDCS*, pages 1595–1598, 2018.

[84] K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez, et al. Tods: An automated time series outlier detection system. In *AAAI*, pages 16060–16062, 2021.

[85] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *NeurIPS*, 2021.

[86] K. Lang. Newsweeder: Learning to filter netnews. In *ICML*, pages 331–339. Elsevier, 1995.

[87] A. Lavin and S. Ahmad. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *2015 IEEE 14th ICML and applications (ICMLA)*, pages 38–44. IEEE, 2015.

[88] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *SDM*, pages 25–36. SIAM, 2003.

[89] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[90] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[91] C. H. Lee and K. Lee. Semi-supervised anomaly detection algorithm based on kl divergence (sad-kl). *arXiv:2203.14539*, 2022.

[92] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 31, 2018.

[93] M.-C. Lee, S. Shekhar, C. Faloutsos, T. N. Hutson, and L. Iasemidis. Gen 2 out: Detecting and ranking generalized anomalies. In *Big Data*, pages 801–811. IEEE, 2021.

[94] M.-C. Lee, Y. Zhao, A. Wang, P. J. Liang, L. Akoglu, V. S. Tseng, and C. Faloutsos. Autoaudit: Mining accounting and time-evolving graphs. In *Big Data*, pages 950–956. IEEE, 2020.

[95] G. Li, Y. Xie, and L. Lin. Weakly supervised salient object detection using image labels. In *AAAI*, volume 32, 2018.

[96] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. Copod: copula-based outlier detection. In *ICDM*, pages 1118–1123. IEEE, 2020.

[97] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *TKDE*, pages 1–1, 2022.

[98] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[99] B. Liu, P. Tan, and J. Zhou. Unsupervised anomaly detection by robust density estimation. In *AAAI*, pages 4101–4108. AAAI Press, 2022.

[100] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *ICDM*, pages 413–422. IEEE, 2008.

[101] K. Liu, Y. Dou, Y. Zhao, X. Ding, X. Hu, R. Zhang, K. Ding, C. Chen, H. Peng, K. Shu, et al. Benchmarking node outlier detection on graphs. *arXiv preprint arXiv:2206.10071*, 2022.

[102] K. Liu, Y. Dou, Y. Zhao, X. Ding, X. Hu, R. Zhang, K. Ding, C. Chen, H. Peng, K. Shu, et al. Pygod: A python library for graph outlier detection. *ArXiv*, 2204.12095, 2022.

[103] S. Liu and M. Hauskrecht. Event outlier detection in continuous time. In *ICML*, pages 6793–6803, 2021.

[104] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. *NeurIPS*, 33:21464–21475, 2020.

[105] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*, 2019.

[106] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K. R. Muller. Explainable deep one-class classification. In *ICLR*, 2020.

[107] W.-Y. Loh. Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, 1, 2011.

[108] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *ArXiv*, 1711.05101, 2017.

[109] M. Q. Ma, Y. Zhao, X. Zhang, and L. Akoglu. A large-scale study on unsupervised outlier model selection: Do internal strategies suffice? *ArXiv*, 2104.01422, 2021.

[110] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *TKDE*, 2021.

[111] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150. Association for Computational Linguistics, 2011.

[112] A. Mahdavi and M. Carvalho. A survey on open set recognition. In *AIKE*, pages 37–44. IEEE, 2021.

[113] D. Malerba, F. Esposito, and G. Semeraro. A further comparison of simplification methods for decision-tree induction. In *Learning from data*, pages 365–374. Springer, 1996.

[114] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

[115] A. Manolache, F. Brad, and E. Burceanu. Date: Detecting anomalies in text via self-supervision of transformers. In *NAACL*, pages 267–277, 2021.

[116] M. Markou and S. Singh. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[117] R. Martinez-Guerra and J. L. Mata-Machuca. *Fault detection and diagnosis in nonlinear systems*. Springer, 2016.

[118] G. W. Milligan. An algorithm for generating artificial test clusters. *Psychometrika*, 50(1):123–127, 1985.

[119] N. Moustafa and J. Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.

[120] N. Mu and J. Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv:1906.02337*, 2019.

[121] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[122] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2019.

[123] G. Pang and C. Aggarwal. Toward explainable deep anomaly detection. In *KDD*, pages 4056–4057, 2021.

[124] G. Pang, L. Cao, and L. Chen. Homophily outlier detection in non-iid categorical data. *Data Mining and Knowledge Discovery*, 35(4):1163–1224, 2021.

[125] G. Pang, L. Cao, L. Chen, and H. Liu. Unsupervised feature selection for outlier detection by modelling hierarchical value-feature couplings. In *ICDM*, pages 410–419. IEEE, 2016.

[126] G. Pang, L. Cao, L. Chen, and H. Liu. Learning homophily couplings from non-iid data for joint feature selection and noise-resilient outlier detection. In *IJCAI*, pages 2585–2591, 2017.

[127] G. Pang, L. Cao, L. Chen, and H. Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *KDD*, pages 2041–2050, 2018.

[128] G. Pang, C. Ding, C. Shen, and A. v. d. Hengel. Explainable deep few-shot anomaly detection with deviation networks. *ArXiv*, 2108.00462, 2021.

[129] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

[130] G. Pang, C. Shen, H. Jin, and A. v. d. Hengel. Deep weakly-supervised anomaly detection. *ArXiv*, 1910.13601, 2019.

[131] G. Pang, C. Shen, and A. van den Hengel. Deep anomaly detection with deviation networks. In *KDD*, pages 353–362, 2019.

[132] J. Paparrizos, Y. Kang, P. Boniol, R. S. Tsay, T. Palpanas, and M. J. Franklin. Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection. *VLDB*, 15(8):1697–1711, 2022.

[133] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the JMLR*, 12:2825–2830, 2011.

[134] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[135] P. Perera, R. Nallapati, and B. Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *CVPR*, pages 2898–2906, 2019.

[136] T. Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

[137] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

[138] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. *NeurIPS*, 31, 2018.

[139] C. Qiu, A. Li, M. Kloft, M. Rudolph, and S. Mandt. Latent outlier exposure for anomaly detection with contaminated data. *ArXiv*, 2202.08088, 2022.

[140] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *ICML*, pages 8703–8714, 2021.

[141] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[142] J. R. Quinlan, P. J. Compton, K. Horn, and L. Lazarus. Inductive knowledge acquisition: a case study. In *Australian Conference on Applications of expert systems*, pages 137–156, 1987.

[143] M. M. Rahman, D. Balakrishnan, D. Murthy, M. Kutlu, and M. Lease. An information retrieval approach to building datasets for hate speech detection. In *NeurIPS*, 2021.

[144] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *SIGMOD*, pages 427–438, 2000.

[145] S. Rayana. ODDS library, 2016.

[146] Q. Rebjock, B. Kurt, T. Januschowski, and L. Callot. Online false discovery rate control for anomaly detection in time series. *NeurIPS*, 34, 2021.

[147] T. Reiss, N. Cohen, L. Bergman, and Y. Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *CVPR*, pages 2806–2814, 2021.

[148] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[149] S. Ruder. An overview of gradient descent optimization algorithms. *ArXiv*, 1609.04747, 2016.

[150] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

[151] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *ICML*, pages 4393–4402, 2018.

[152] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *ICLR*. OpenReview.net, 2020.

[153] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, and M. Kloft. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *ACL*, pages 4061–4071, 2019.

[154] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. In *CVPR*, pages 3379–3388, 2018.

[155] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban, and M. Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv:2110.14051*, 2021.

[156] R. Schirrmeister, Y. Zhou, T. Ball, and D. Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *NeurIPS*, 33:21038–21049, 2020.

[157] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588. Citeseer, 1999.

[158] V. Sehwag, M. Chiang, and P. Mittal. Ssd: A unified framework for self-supervised outlier detection. In *ICLR*, 2020.

[159] S. Shekhar, N. Shah, and L. Akoglu. Fairod: Fairness-aware outlier detection. In *AIES*, pages 210–220, 2021.

[160] L. Shen, Z. Li, and J. Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *NeurIPS*, 33:13016–13026, 2020.

[161] T. Shenkar and L. Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2021.

[162] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering, 2003.

[163] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10(3):262–266, 1989.

[164] J. Soenen, E. Van Wolputte, L. Perini, V. Vercruyssen, W. Meert, J. Davis, and H. Blockeel. The effect of hyperparameter tuning on the comparative evaluation of unsupervised anomaly detection methods. In *Proceedings of the KDD'21 Workshop on Outlier Detection and Description*, pages 1–9, 2021.

[165] H. Song, P. Li, and H. Liu. Deep clustering based fair outlier detection. In *KDD*, pages 1481–1489, 2021.

[166] G. Steinbuss and K. Böhm. Benchmarking unsupervised outlier detection with realistic synthetic data. *TKDD*, 15(4):1–20, 2021.

[167] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *PAKDD*, pages 535–548. Springer, 2002.

[168] B. Tian, Q. Su, and J. Yin. Anomaly detection by leveraging incomplete anomalous knowledge with anomaly-aware bidirectional gans. *ArXiv*, 2204.13335, 2022.

[169] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[170] S. Vargaftik, I. Keslassy, A. Orda, and Y. Ben-Itzhak. Rade: Resource-efficient supervised anomaly detection using decision tree-based ensemble methods. *Machine Learning*, 110(10):2835–2866, 2021.

[171] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[172] Z. Wang, B. Dai, D. Wipf, and J. Zhu. Further analysis of outlier detection with deep generative models. *NeurIPS*, 33:8982–8992, 2020.

[173] W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the national academy of sciences*, 87(23):9193–9196, 1990.

[174] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[175] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[176] K. S. Woods, J. L. Solka, C. E. Priebe, W. P. Kegelmeyer Jr, C. C. Doss, and K. W. Bowyer. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. In *State of The Art in Digital Mammographic Image Analysis*, pages 213–231. World Scientific, 1994.

[177] B. Wu, J. Chen, D. Cai, X. He, and Q. Gu. Do wider neural networks really help adversarial robustness? *NeurIPS*, 34, 2021.

[178] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

[179] Z. Xiao, Q. Yan, and Y. Amit. Do we really need to learn representations from in-domain data for outlier detection? *ArXiv*, 2105.09270, 2021.

[180] H. Xu, Y. Wang, S. Jian, Z. Huang, Y. Wang, N. Liu, and F. Li. Beyond outlier detection: Outlier interpretation by attention-guided triplet deviation network. In *WWW*, pages 1328–1339, 2021.

[181] Y. Xu, J. Ding, L. Zhang, and S. Zhou. Dp-ssl: Towards robust semi-supervised learning with a few labeled samples. *NeurIPS*, 34, 2021.

[182] J. Yang, P. Wang, D. Zou, Z. Zhou, K. Ding, W. Peng, H. Wang, G. Chen, B. Li, Y. Sun, et al. Openood: Benchmarking generalized out-of-distribution detection. 2022.

[183] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *arXiv:2110.11334*, 2021.

[184] L. Yang, Z. Zhang, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, M.-H. Yang, and B. Cui. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[185] Z. Yang, I. S. Bozchalooi, and E. Darve. Anomaly detection with domain adaptation. *arXiv:2006.03689*, 2020.

[186] W. Yu, J. Li, M. Z. A. Bhuiyan, R. Zhang, and J. Huai. Ring: Real-time emerging anomaly monitoring system over text streams. *IEEE Transactions on Big Data*, 5(4):506–519, 2017.

[187] M. D. Zeiler. Adadelta: an adaptive learning rate method. *ArXiv*, 1212.5701, 2012.

[188] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar. Adversarially learned anomaly detection. In *ICDM*, pages 727–736. IEEE, 2018.

[189] D. Zha, K.-H. Lai, M. Wan, and X. Hu. Meta-aad: Active anomaly detection with deep reinforcement learning. In *ICDM*, pages 771–780. IEEE, 2020.

[190] H. Zhang and I. Davidson. Towards fair deep anomaly detection. In *FAccT*, pages 138–148, 2021.

[191] S. Zhang, V. Ursekar, and L. Akoglu. Sparx: Distributed outlier detection at scale. *KDD*, 2022.

[192] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *NIPS*, 28, 2015.

[193] J. Zhao, X. Liu, Q. Yan, B. Li, M. Shao, and H. Peng. Multi-attributed heterogeneous graph convolutional network for bot detection. *Information Sciences*, 537:380–393, 2020.

[194] Y. Zhao and L. Akoglu. Towards unsupervised hpo for outlier detection. *arXiv preprint arXiv:2208.11727*, 2022.

[195] Y. Zhao and M. K. Hryniewicki. Xgbod: improving supervised outlier detection with unsupervised representation learning. In *IJCNN*, pages 1–8. IEEE, 2018.

[196] Y. Zhao, X. Hu, C. Cheng, C. Wang, C. Wan, W. Wang, J. Yang, H. Bai, Z. Li, C. Xiao, et al. Suod: Accelerating large-scale unsupervised heterogeneous outlier detection. *MLSys*, 3:463–478, 2021.

[197] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, and Z. Li. Lscp: Locally selective combination in parallel outlier ensembles. In *SDM*, pages 585–593. SIAM, 2019.

[198] Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. *JMLR*, 20:1–7, 2019.

[199] Y. Zhao, R. Rossi, and L. Akoglu. Automatic unsupervised outlier model selection. *NeurIPS*, 34, 2021.

[200] Y. Zhao, S. Zhang, and L. Akoglu. Toward unsupervised outlier model selection. In *IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022.

[201] Y. Zhao, G. Zheng, S. Mukherjee, R. McCann, and A. Awadallah. Admoe: Anomaly detection with mixture-of-experts from noisy labels. *arXiv preprint arXiv:2208.11290*, 2022.

[202] G. Zheng, A. H. Awadallah, and S. Dumais. Meta label correction for noisy label learning. *AAAI*, 2021.

[203] Y. Zheng, X. Wang, Y. Qi, W. Li, and L. Wu. Benchmarking unsupervised anomaly detection and localization. *ArXiv*, 2205.14852, 2022.

[204] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan. Learning placeholders for open-set recognition. In *CVPR*, pages 4401–4410, 2021.

[205] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu. Feature encoding with autoencoders for weakly supervised anomaly detection. *TNNLS*, 2021.

[206] Z.-H. Zhou. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.*, 5(1):44–53, 2018.

[207] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018.