INVITED PAPER

A Unifying Review of Deep and Shallow Anomaly Detection

This article deals with application of deep learning techniques to anomaly detection. Furthermore, connections between classic "shallow" and novel deep approaches are established, and it is shown how this relation might cross-fertilize or extend both directions.

By Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, *Member IEEE*, Marius Kloft, *Senior Member IEEE*, Thomas G. Dietterich, *Member IEEE*, and Klaus-Robert Müller, *Member IEEE*

ABSTRACT | Deep learning approaches to anomaly detection (AD) have recently improved the state of the art in detection performance on complex data sets, such as large collections of images or text. These results have sparked a renewed interest in the AD problem and led to the introduction of a great variety of new methods. With the emergence of numerous such methods, including approaches based on generative models, one-class classification, and reconstruction, there is a growing need to bring methods of this field into a systematic and unified perspective. In this review, we aim to identify the common underlying principles and the assumptions that are often made implicitly by various methods. In particular, we draw connections between classic "shallow" and novel deep approaches and show how this relation might cross-fertilize or extend both directions. We further provide an empirical assessment of major existing methods that are enriched by the use of recent explainability techniques and present specific worked-through examples together with practical advice. Finally, we outline critical open challenges and identify specific paths for future research in AD.

KEYWORDS | Anomaly detection (AD); deep learning; explainable artificial intelligence; interpretability; kernel methods; neural networks; novelty detection; one-class classification; outlier detection; out-of-distribution (OOD) detection; unsupervised learning.

NOMENCLATURE

AD	Anomaly detection.
AE	Autoencoder.
AP	Average precision.
AAE	Adversarial AE.
AUPRC	Area under the precision–recall curve.
AUROC	Area under the ROC curve.
CAE	Contrastive AE.
DAE	Denoising AE.
DGM	Deep generative model.
DSVDD	Deep support vector data description.

Manuscript received September 12, 2020; revised December 13, 2020; accepted January 7, 2021. Date of publication February 4, 2021; date of current version April 30, 2021. The work of Lukas Ruff was supported by the German Federal Ministry of Education and Research (BMBF) through the project ALICE III under Grant 01IS18049B. The work of Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller was supported in part by the Berlin Institute for the Foundations of Learning and Data (BIFOLD) sponsored by the BMBF. The work of Marius Kloft was supported in part by the German Research Foundation (DFG) under Award KL 2698/2-1 and in part by the BMBF under Award 01IS18051A and Award 031B0770E. The work of Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller was supported in part by the BMBF for the Berlin Center for Machine Learning (01IS18037A-I) under Grant 01IS14013A-E and Grant 031L0207A-D. The work of Klaus-Robert Müller was supported in part by BMBF under Grant 01GQ1115 and Grant 01GQ0850, in part by DFG under Grant Math+, EXC 2046/1, Project ID 390685689, in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea Government under Grant 2017-0-00451 (Development of BCI Based Brain and Cognitive Computing Technology for Recognizing User's Intentions using Deep Learning), and in part by the Korea Government through the Artificial Intelligence Graduate School Program, Korea University, under Grant 2019-0-00079. The work of Thomas G. Dietterich was supported by the U.S. Defense Advanced Research Projects Agency (DARPA) under Contract HR001119C0112, Contract FA8750-19-C-0092, and Contract HR001120C0022. *(Corresponding authors: Marius Kloft; Thomas G. Dietterich; Klaus-Robert Müller.)*

Lukas Ruff, **Jacob R. Kauffmann**, **Robert A. Vandermeulen**, and **Grégoire Montavon** are with the ML Group, Technische Universität Berlin, 10587 Berlin, Germany.

Wojciech Samek is with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany.

Marius Kloft is with the Department of Computer Science, Technische Universität Kaiserslautern, 67653 Kaiserslautern, Germany (e-mail: kloft@cs.uni-kl.de).

Thomas G. Dietterich is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA (e-mail: tgd@cs.orst.edu).

Klaus-Robert Müller is with the Brain Team, Google Research, 10117 Berlin, Germany, with the ML Group, Technische Universität Berlin, 10587 Berlin, Germany, with the Department of Artificial Intelligence, Korea University, Seoul 136-713, South Korea, and also with the Max Planck Institute for Informatics, 66123 Saarbrücken, Germany (e-mail: klaus-robert.mueller@tu-berlin.de).

Digital Object Identifier 10.1109/JPROC.2021.3052449

DSAD	Deep semisupervised AD.
EBM	Energy-based model.
ELBO	Evidence lower bound.
GAN	Generative adversarial network.
GMM	Gaussian mixture model.
GT	Geometric transformation.
iForest	Isolation forest.
KDE	Kernel density estimation.
k-NN	k-nearest neighbors.
kPCA	Kernel principal component analysis.
LOF	Local outlier factor.
LPUE	Learning from positive and unlabeled examples.
LSTM	Long short-term memory.
MCMC	Markov chain Monte Carlo.
MCD	Minimum covariance determinant.
MVE	Minimum volume ellipsoid.
OOD	Out-of-distribution.
OE	Outlier exposure.
OC-NN	One-class neural network.
OC-SVM	One-class support vector machine.
pPCA	Probabilistic principal component analysis.
PCA	Principal component analysis.
pdf	Probability density function.
PSD	Positive semidefinite.
RBF	Radial basis function.
RKHS	Reproducing kernel Hilbert space.
rPCA	Robust PCA.
SGD	Stochastic gradient descent.
SGLD	Stochastic gradient Langevin dynamics.
SSAD	Semisupervised AD.
SVDD	Support vector data description.
VAE	Variational AE.
VQ	Vector quantization.
XAI	Explainable AI.

I. INTRODUCTION

An anomaly is an observation that deviates considerably from some concept of normality. Also known as outlier or novelty, such an observation may be termed unusual, irregular, atypical, inconsistent, unexpected, rare, erroneous, faulty, fraudulent, malicious, unnatural, or simply strange—depending on the situation. AD (or outlier detection or novelty detection) is the research area that studies the detection of such anomalous observations through methods, models, and algorithms based on data. Classic approaches to AD include PCA [1]–[5], the OC-SVM [6], SVDD [7], nearest neighbor algorithms [8]–[10], and KDE [11], [12].

What the above methods have in common is that they are all unsupervised, which constitutes the predominant approach to AD. This is because, in standard AD settings, labeled anomalous data are often nonexistent. When available, it is usually insufficient to fully characterize all notions of anomalousness. This typically makes a supervised approach ineffective. Instead, a central idea in AD is to learn a model of normality from normal data in an unsupervised manner so that anomalies become detectable through deviations from the model.

The study of AD has a long history and spans multiple disciplines, including engineering, machine learning, data mining, and statistics. While the first formal definitions of so-called "discordant observations" date back to the 19th century [13], the problem of AD has likely been studied informally even earlier since anomalies are phenomena that naturally occur in diverse academic disciplines, such as medicine and the natural sciences. Anomalous data may be useless, for example, when caused by measurement errors, or maybe extremely informative and hold the key to new insights, such as very long-surviving cancer patients. Kuhn [14] claimed that persistent anomalies drive scientific revolutions (see [14, Section VI]).

AD today has numerous applications across a variety of domains. Examples include intrusion detection in cybersecurity [15]–[20], fraud detection in finance, insurance, healthcare, and telecommunication [21]–[27], industrial fault and damage detection [28]–[36], the monitoring of infrastructure [37], [38] and stock markets [39], [40], acoustic novelty detection [41]–[45], medical diagnosis [46]–[60] and disease outbreak detection [61], [62], event detection in the earth sciences [63]–[68], and scientific discovery in chemistry [69], [70], bioinformatics [71], genetics [72], [73], physics [74], [75], and astronomy [76]–[79]. The data available in these domains is continually growing in size. It is also expanding to include complex data types, such as images, videos, audios, text, graphs, multivariate time series, and biological sequences, among others. For applications to be successful in such complex and high-dimensional data, a meaningful representation of the data is crucial [80].

Deep learning [81]–[83] follows the idea of learning effective representations from the data itself by training flexible, multilayered ("deep") neural networks and has greatly improved the state of the art in many applications that involve complex data types. Deep neural networks provide the most successful solutions for many tasks in domains, such as computer vision [84]–[93], speech recognition [94]–[103], or natural language processing [104]–[113] and have contributed to the sciences [114]–[123]. Methods based on deep neural networks are able to exploit the hierarchical or latent structure that is often inherent to data through their multilayered, distributed feature representations. Advances in parallel computation, SGD optimization, and automated differentiation make it possible to apply deep learning at scale using large data sets.

Recently, there has been a surge of interest in developing deep learning approaches for AD. This is motivated by a lack of effective methods for AD tasks that involve complex data, for instance, cancer detection from multigigapixel whole-slide images in histopathology [124]. As in other adoptions of deep learning, the goal of deep AD is to mitigate the burden of manual feature engineering and to enable effective, scalable solutions. However, unlike supervised deep learning, it is less clear what useful representation learning objectives for deep AD are, due to the mostly unsupervised nature of the problem.

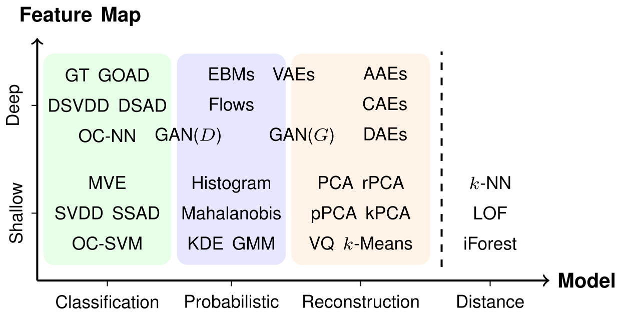


Fig. 1. ***AD approaches arranged in the plane spanned by two major components (model and feature map) of our unifying view. Based on shared principles, we distinguish one-class classification, probabilistic models, and reconstruction models as the three main groups of approaches that all formulate shallow and deep models (see Nomenclature for a list of abbreviations). These three groups are complemented by purely distance-based methods. Besides model and feature map, we identify loss, regularization, and inference mode as other important modeling components of the AD problem.***

The major approaches to deep AD include deep AE variants [44], [51], [54], [125]–[135], deep one-class classification [136]–[145], methods based on DGMs, such as GANs [50], [56], [146]–[151], and recent self-supervised methods [152]–[156]. In comparison to traditional AD methods, where a feature representation is fixed *a priori* (e.g., via a kernel feature map), these approaches aim to learn a feature map of the data $\phi_\omega : \boldsymbol{x} \mapsto \phi_\omega(\boldsymbol{x})$, a deep neural network parameterized with weights ω, as part of their learning objective.

Due to the long history and diversity of AD research, there exists a wealth of review and survey literature [157]–[176] and books [177]–[179] on the topic. Some very recent surveys focus specifically on deep AD [180]–[182]. However, an integrated treatment of deep learning methods in the overall context of AD research—in particular, its kernel-based learning part [6], [7], [183]—is still missing.

In this review article, we aim to fill this gap by presenting a unifying view that connects traditional shallow and novel deep learning approaches. We will summarize recent exciting developments, present different classes of AD methods, provide theoretical insights, and highlight the current best practices when applying AD. Fig. 1 gives an overview of the categorization of AD methods within our unifying view. Note, finally, that we do not attempt an encyclopedic treatment of all available AD literature; rather, we present a slightly biased point of view (drawing from our own work on the subject), illustrating the main topics, and provide ample reference to related work for further reading.

II. AN INTRODUCTION TO ANOMALY DETECTION

A. Why Should We Care About Anomaly Detection?

Though we may not realize it, AD is part of our daily life. Operating mostly unnoticed, AD algorithms are continuously monitoring our credit card payments, our login behaviors, and companies' communication networks. If these algorithms detect an abnormally expensive purchase made on our credit card, several unsuccessful login attempts made from an alien device in a distant country, or unusual FTP requests made to our computer, they will issue an alarm. While warnings, such as "someone is trying to login to your account," can be annoying when you are on a business trip abroad and just want to check your e-mails from the hotel computer, the ability to detect such anomalous patterns is vital for a large number of today's applications and services, and even small improvements in AD can lead to immense monetary savings.[1]

In addition, the ability to detect anomalies is also an important ingredient in ensuring fail-safe and robust design of deep learning-based systems, for instance, in medical applications or autonomous driving. Various international standardization initiatives have been launched toward this goal (e.g., ITU/WHO FG-AI4H, ISO/IEC CD TR 24029-1, or IEEE P7009).

Despite its importance, discovering a reliable distinction between "normal" and "anomalous" events is a challenging task. First, the variability within normal data can be very large, resulting in misclassifying normal samples as being anomalous (type I error) or not identifying the anomalous ones (type II error). Especially in biological or biomedical data sets, the variability between the normal data (e.g., person-to-person variability) is often as large or even larger than the distance to anomalous samples (e.g., patients). Preprocessing, normalization, and feature selection are potential means to reduce this variability and improve detectability [179], [184], [185]. If such steps are neglected, the features with wide value ranges, noise, or irrelevant features can dominate distance computations and "mask" anomalies [165] (see VIII-A). Second, anomalous events are often very rare, which results in highly imbalanced training data sets. Even worse, in most cases, the data set is unlabeled so that it remains unclear which data points are anomalies and why. Hence, AD reduces to an unsupervised learning task with the goal to learn a valid model of the majority of data points. Finally, anomalies themselves can be very diverse so that it becomes difficult to learn a complete model for them. Likewise, the solution is again to learn a model for the normal samples and treat deviations from it as anomalies. However, this approach can be problematic if the distribution of the normal data changes (nonstationarity), either intrinsically or due to environmental changes (e.g., lighting conditions and recording devices from different manufacturers).

As exemplified and discussed above, we note that AD has broad practical relevance and impact. Moreover, (accidentally) detecting the unknown unknowns [186] is a strong driving force in the sciences. If applied in the sciences, AD can help us to identify new, previously unknown

[1] In 2019, U.K.'s online banking fraud has been estimated to be 111.8 million GBP (source: https://www.statista.com/).

patterns in data, which can lead to novel scientific insights and hypotheses.

B. Formal Definition of Anomaly Detection

In the following, we formally introduce the AD problem. We first define in probabilistic terms what an anomaly is, explain what types of anomalies there are, and delineate the subtle differences between an anomaly, an outlier, and a novelty. Finally, we present a fundamental principle in AD—the so-called concentration assumption—and give a theoretical problem formulation that corresponds to density level set estimation.

1) What Is an Anomaly?: We opened this review with the following definition:

> An anomaly is an observation that deviates considerably from some concept of normality.

To formalize this definition, we here specify two aspects more precisely: a "concept of normality" and what "deviates considerably" signifies. Following many previous authors [13], [177], [187]–[189], we rely on probability theory.

Let $\mathcal{X} \subseteq \mathbb{R}^D$ be the data space given by some task or application. We define a concept of normality as the distribution $\mathbb{P}^+$ on $\mathcal{X}$ that is the ground-truth law of normal behavior in a given task or application. An observation that deviates considerably from such a law of normality—an anomaly—is, then, a data point $\boldsymbol{x} \in \mathcal{X}$ (or set of points) that lies in a low probability region under $\mathbb{P}^+$. Assuming that $\mathbb{P}^+$ has a corresponding pdf $p^+(\boldsymbol{x})$, we can define a set of anomalies as

$$\mathcal{A} = \{\boldsymbol{x} \in \mathcal{X} \mid p^+(\boldsymbol{x}) \leq \tau\}, \quad \tau \geq 0 \tag{1}$$

where τ is some threshold such that the probability of $\mathcal{A}$ under $\mathbb{P}^+$ is "sufficiently small" that we will specify further in the following.

2) Types of Anomalies: Various types of anomalies have been identified in the literature [161], [179]. These include point anomalies, conditional or contextual anomalies [169], [171], [191]–[195], and group or collective anomalies [146], [193], [196]–[199]. We extend these three established types by further adding low-level sensory anomalies and high-level semantic anomalies [200], a distinction that is particularly relevant for choosing between deep and shallow feature maps.

A point anomaly is an individual anomalous data point $\boldsymbol{x} \in \mathcal{A}$, for example, an illegal transaction in fraud detection or an image of a damaged product in manufacturing. This is arguably the most commonly studied type in AD research.

A conditional or contextual anomaly is a data instance that is anomalous in a specific context, such as time, space, or the connections in a graph. A price of \$1 per Apple Inc. stock might have been normal before 1997 but, as of today (2021), would be an anomaly. A mean daily temperature below freezing point would be an anomaly in the Amazon rainforest but not in the Antarctic desert. For this anomaly type, the normal law $\mathbb{P}^+$ is more precisely a conditional distribution $\mathbb{P}^+ \equiv \mathbb{P}^+_{X|T}$ with conditional pdf $p^+(\boldsymbol{x} \,|\, t)$ that depends on some contextual variable T. Time-series anomalies [169], [195], [201]–[204] are the most prominent example of contextual anomalies. Other examples include spatial [205], [206], spatiotemporal [192], or graph-based [171], [207], [208] anomalies.

A group or collective anomaly is a set of related or dependent points $\{\boldsymbol{x}_j \in \mathcal{X} \mid j \in J\}$ that are anomalous, where $J \subseteq \mathbb{N}$ is an index set that captures some relation or dependence. A cluster of anomalies, such as similar or related network attacks in cybersecurity, forms a collective anomaly, for instance [18], [208], [209]. Often, collective anomalies are also contextual, such as anomalous time (sub)series or biological (sub)sequences, for example, some series or sequence $\{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1}\}$ of length $s \in \mathbb{N}$. It is important to note that although each individual point $\boldsymbol{x}_j$ in such a series or sequence might be normal under the time-integrated marginal $p^+(\boldsymbol{x}) = \int p^+(\boldsymbol{x}, t)\,\mathrm{d}t$ or under the sequence-integrated, time-conditional marginal $p^+(\boldsymbol{x} \,|\, t)$ given by

$$\int \cdots \int p^+(\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1} \,|\, t)\,\mathrm{d}\boldsymbol{x}_t \cdots \mathrm{d}\boldsymbol{x}_{j-1}\,\mathrm{d}\boldsymbol{x}_{j+1} \cdots \mathrm{d}\boldsymbol{x}_{t+s-1}$$

the full series or sequence $\{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1}\}$ can be anomalous under the joint conditional density $p^+(\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+s-1} \,|\, t)$, which properly describes the distribution of the collective series or sequences.

In the wake of deep learning, a distinction between low-level sensory anomalies and high-level semantic anomalies [200] has become important. Low and high here refer to the level in the feature hierarchy of some hierarchical distribution, for instance, the hierarchy from pixel-level features, such as edges and textures to high-level objects and scenes in images or the hierarchy from individual characters and words to semantic concepts and topics in text. It is commonly assumed that data with such a hierarchical structure is generated from some semantic latent variables Z and Y that describe higher level factors of variation Z (e.g., the shape, size, or orientation of an object) and concepts Y (e.g., the object class identity) [80], [210]. We can express this via a law of normality with conditional pdf $p^+(\boldsymbol{x} \,|\, \boldsymbol{z}, y)$, where we usually assume Z to be continuous and Y to be discrete. Low-level anomalies could be texture defects or artifacts in images, or character typos in words. In comparison, semantic anomalies could be images of objects from nonnormal classes [200], for instance, or misposted reviews and news articles [140]. Note that semantic anomalies can be very close to normal instances in the raw feature space $\mathcal{X}$. For example, a dog with a fur texture and color similar to that of some cat can be more similar in raw pixel space than various cat breeds among themselves (see Fig. 2). Similarly, low-level

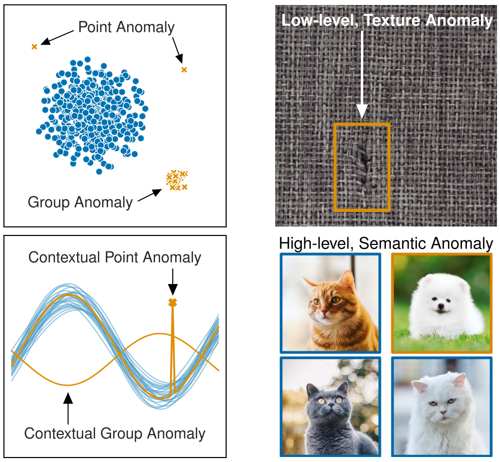


Fig. 2. ***Illustration of the types of anomalies: a point anomaly is a single anomalous point. A contextual point anomaly occurs if a point deviates in its local context, here a spike in an otherwise normal time series. A group anomaly can be a cluster of anomalies or some series of related points that are anomalous under the joint series distribution (contextual group anomaly). Note that both contextual anomalies have values that fall into the global (time-integrated) range of normal values. A low-level sensory anomaly deviates from the low-level features, here a cut in the fabric texture of a carpet [190]. A semantic anomaly deviates in high-level factors of variation or semantic concepts, here a dog among the normal class of cats. Note that the white cat is more similar to the dog than to the other cats in low-level pixel space.***

background statistics can also result in a high similarity in raw pixel space even when objects in the foreground are completely different [200]. Detecting semantic anomalies is, thus, innately tied to finding a semantic feature representation (e.g., extracting the semantic features of cats, such as whiskers, slit pupils, and triangular snout), which is an inherently difficult task in an unsupervised setting [210].

3) Anomaly, Outlier, or Novelty?: Some studies make a concrete (albeit subtle) distinction between what is an anomaly, an outlier, or a novelty. While all three refer to instances from low probability regions under $\mathbb{P}^{+}$ (i.e., are elements of $\mathcal{A}$), an anomaly is often characterized as being an instance from a distinct distribution other than $\mathbb{P}^{+}$ (e.g., when anomalies are generated by a different process than the normal points), an outlier as being a rare or low-probability instance from $\mathbb{P}^{+}$, and a novelty as being an instance from some new region or mode of an evolving, nonstationary $\mathbb{P}^{+}$. Under the distribution $\mathbb{P}^{+}$ of cats, for instance, a dog would be an anomaly, a rare breed of cats, such as the LaPerm, would be an outlier, and a new breed of cats would be a novelty. Such a distinction between anomaly, outlier, and novelty may reflect slightly different objectives in an application: while anomalies are often the data points of interest (e.g., a long-term survivor of a disease), outliers are frequently regarded as "noise" or "measurement error" that should be removed in a data preprocessing step ("outlier removal"), and novelties are new observations that require models to be updated to the "new normal." The methods for detecting points from low probability regions, whether termed "anomaly," "outlier," or "novelty," are essentially the same, however. For this reason, we make no distinction between these terms and call any instance $\boldsymbol{x} \in \mathcal{A}$ an "anomaly."

4) Concentration Assumption: While, in most situations, the data space $\mathcal{X} \subseteq \mathbb{R}^D$ is unbounded, a fundamental assumption in AD is that the region where the normal data lives can be bounded. That is, there exists some threshold $\tau \geq 0$ such that

$$\mathcal{X} \setminus \mathcal{A} = \{\boldsymbol{x} \in \mathcal{X} \mid p^{+}(\boldsymbol{x}) > \tau\} \qquad (2)$$

is nonempty and small (typically, in the Lebesgue-measure sense, which is the ordinary notion of volume in D-dimensional space). This is known as the so-called concentration or cluster assumption [211]–[213]. Note that the concentration assumption does not imply that the full support $\mathrm{supp}(p^{+}) = \{\boldsymbol{x} \in \mathcal{X} \mid p^{+}(\boldsymbol{x}) > 0\}$ of the normal law $\mathbb{P}^{+}$ must be bounded; only that some high-density subset of the support is bounded. A standard univariate Gaussian's support is the full real axis, for example, but approximately 95% of its probability mass is contained in the interval $[-1.96, 1.96]$. In contrast, the set of anomalies $\mathcal{A}$ need not be concentrated and can be unbounded.

5) Density Level Set Estimation: A law of normality $\mathbb{P}^{+}$ is only known in a few application settings, such as for certain laws of physics. Sometimes, a concept of normality might also be user-specified (as in juridical laws). In most cases, however, the ground-truth law of normality $\mathbb{P}^{+}$ is unknown because the underlying process is too complex. For this reason, we must estimate $\mathbb{P}^{+}$ from data.

Let $\mathbb{P}$ be the ground-truth data-generating distribution on data space $\mathcal{X} \subseteq \mathbb{R}^D$ with corresponding density $p(\boldsymbol{x})$, that is, the distribution that generates the observed data. For now, we assume that this data-generating distribution exactly matches the normal data distribution, that is, $\mathbb{P} \equiv \mathbb{P}^{+}$ and $p \equiv p^{+}$. This assumption is often invalid in practice, of course, as the data-generating process might be subject to noise or contamination, as we will discuss in Section II-C.

Given data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ generated by $\mathbb{P}$ (usually assumed to be drawn from i.i.d. random variables following $\mathbb{P}$), the goal of AD is to learn a model that allows us to predict whether a new test instance $\tilde{\boldsymbol{x}} \in \mathcal{X}$ is an anomaly or not, that is, whether $\tilde{\boldsymbol{x}} \in \mathcal{A}$. Thus, the AD objective is to (explicitly or implicitly) estimate the low-density regions (or equivalently high-density regions) in data space $\mathcal{X}$ under the normal law $\mathbb{P}^{+}$. We can formally express this objective as the problem of density level set estimation [214]–[217], which is equivalent to minimum volume set estimation [218]–[220] for the special case of density-based sets. The density level set of $\mathbb{P}$ for some

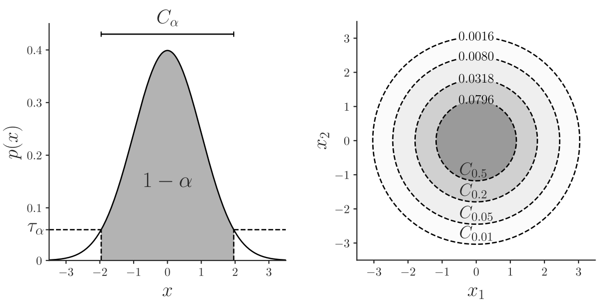


Fig. 3. ***Illustration of the α-density level sets C_α with threshold τ_α for a univariate (left) and bivariate (right) standard Gaussian distribution.***

threshold $\tau \geq 0$ is given by $C = \{\boldsymbol{x} \in \mathcal{X} \,|\, p(\boldsymbol{x}) > \tau\}$. For some fixed level $\alpha \in [0,1]$, the α-density level set C_α of distribution $\mathbb{P}$ is then defined as the smallest density level set C that has a probability of at least $1-\alpha$ under $\mathbb{P}$, that is,

$$\begin{aligned} C_\alpha &= \operatorname*{arginf}_C \{\lambda(C) \mid \mathbb{P}(C) \geq 1-\alpha\} \\ &= \{\boldsymbol{x} \in \mathcal{X} \,|\, p(\boldsymbol{x}) > \tau_\alpha\} \end{aligned} \tag{3}$$

where $\tau_\alpha \geq 0$ denotes the corresponding threshold and λ is typically the Lebesgue measure. The extreme cases of $\alpha = 0$ and $\alpha \to 1$ result in the full support $C_0 = \{\boldsymbol{x} \in \mathcal{X} \,|\, p(\boldsymbol{x}) > 0\} = \operatorname{supp}(p)$ and the most likely modes $\operatorname{argmax}_{\boldsymbol{x}} p(\boldsymbol{x})$ of $\mathbb{P}$, respectively. If the aforementioned concentration assumption holds, there always exists some level α such that a corresponding level set C_α exists and can be bounded. Fig. 3 illustrates some density level sets for the case that $\mathbb{P}$ is the familiar standard Gaussian distribution. Given a level set C_α, we can define a corresponding threshold anomaly detector $c_\alpha : \mathcal{X} \to \{\pm 1\}$ as

$$c_\alpha(\boldsymbol{x}) = \begin{cases} +1, & \text{if } \boldsymbol{x} \in C_\alpha \\ -1, & \text{if } \boldsymbol{x} \notin C_\alpha. \end{cases} \tag{4}$$

6) Density Estimation for Level Set Estimation: An obvious approach to density level set estimation is through density estimation. Given some estimated density model $\hat{p}(\boldsymbol{x}) = \hat{p}(\boldsymbol{x}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \approx p(\boldsymbol{x})$ and some target level $\alpha \in [0,1]$, one can estimate a corresponding threshold $\hat{\tau}_\alpha$ via the empirical p-value function

$$\hat{\tau}_\alpha = \inf_\tau \left\{ \tau \geq 0 \;\middle|\; \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{[0,\hat{p}(\boldsymbol{x}_i))}(\tau) \geq 1-\alpha \right\} \tag{5}$$

where $\mathbb{1}_A(\cdot)$ denotes the indicator function for some set A. Using $\hat{\tau}_\alpha$ and $\hat{p}(\boldsymbol{x})$ in (3) yields the plug-in density level set estimator $\hat{C}_\alpha$, which can be used in (4) to obtain the plug-in threshold detector $\hat{c}_\alpha(\boldsymbol{x})$. Note, however, that density estimation is generally the most costly approach to density level set estimation (in terms of samples required) since estimating the full density is equivalent to first estimating the entire family of level sets $\{C_\alpha \,|\, \alpha \in [0,1]\}$ from which the desired level set for some fixed $\alpha \in [0,1]$ is then selected [221], [222]. If there are insufficient samples, this density estimate can be biased. This has also motivated the development of one-class classification methods that aim to estimate a collection [222] or single-level sets [6], [7], [223], [224] directly, which we will explain in Section IV in more detail.

7) Threshold Versus Score: The previous approach to level set estimation through density estimation is relatively costly, yet results in a more informative model that can rank inliers and anomalies according to their estimated density. In comparison, a pure threshold detector as in (4) only yields a binary prediction. Menon and Williamson [222] proposed a compromise by learning a density outside the level set boundary. Many AD methods also target some strictly increasing transformation $T : [0,\infty) \to \mathbb{R}$ of the density for estimating a model (e.g., log-likelihood instead of likelihood). The resulting target $T(p(\boldsymbol{x}))$ is usually no longer a proper density but still preserves the density ranking [225], [226]. An anomaly score $s : \mathcal{X} \to \mathbb{R}$ can then be defined by using an additional order-reversing transformation, for example, $s(\boldsymbol{x}) = -T(p(\boldsymbol{x}))$ (e.g., negative log-likelihood) so that high scores reflect low-density values, and vice versa. Having such a score that indicates the “degree of anomalousness” is important in many AD applications. As for the density in (5), of course, we can always derive a threshold from the empirical distribution of anomaly scores if needed.

8) Selecting a Level α: As we will show, there are many degrees of freedom when attacking the AD problem, which inevitably requires making various modeling assumptions and choices. Setting the level α is one of these choices and depends on the specific application. When the value of α increases, the anomaly detector focuses only on the most likely regions of $\mathbb{P}$. Such a detector can be desirable in applications where missed anomalies are costly (e.g., in medical diagnosis or fraud detection). On the other hand, a large α will result in high false alarm rates, which can be undesirable in online settings where lots of data is generated (e.g., in monitoring tasks). We provide a practical example for selecting α in Section VIII. Choosing α also involves further assumptions about the data-generating process $\mathbb{P}$, which we have assumed here to match the normal data distribution $\mathbb{P}^+$. In Section II-C, we discuss the data settings that can occur in AD that may alter this assumption.

C. Data Set Settings and Data Properties

The data set settings (e.g., unsupervised or semisupervised) and data properties (e.g., type or dimensionality) that occur in real-world AD problems can be diverse.

We here characterize these settings, which may range from the standard unsupervised to semisupervised and supervised settings, and list further data properties that are relevant for modeling an AD problem. However, before we elaborate on these, we first observe that the assumptions made about the distribution of anomalies (often implicitly) are also crucial to the problem.

1) Distribution of Anomalies?: Let $\mathbb{P}^-$ denote the ground-truth anomaly distribution and assume that it exists on $\mathcal{X} \subseteq \mathbb{R}^D$. As mentioned above, the common concentration assumption implies that some high-density regions of the normal data distribution are concentrated, whereas anomalies are assumed to be not concentrated [211], [212]. This assumption may be modeled by an anomaly distribution $\mathbb{P}^-$ that is a uniform distribution over the (bounded[2]) data space $\mathcal{X}$ [224]. Some well-known unsupervised methods, such as KDE [12] or the OC-SVM [6], implicitly make this assumption that $\mathbb{P}^-$ follows a uniform distribution that can be interpreted as a default uninformative prior on the anomalous distribution [212]. This prior assumes that there are no anomalous modes and that anomalies are equally likely to occur over the valid data space $\mathcal{X}$. Semisupervised or supervised AD approaches often depart from this uninformed prior and try to make a more informed *a priori* assumption about the anomalous distribution $\mathbb{P}^-$ [212]. If faithful to $\mathbb{P}^-$, such a model based on a more informed anomaly prior can achieve better detection performance. Modeling anomalous modes can also be beneficial in certain applications, for example, for typical failure modes in industrial machines or known disorders in medical diagnosis. We remark that these prior assumptions about the anomaly distribution $\mathbb{P}^-$ are often expressed only implicitly in the literature though such assumptions are critical to an AD model.

2) Unsupervised Setting: The unsupervised AD setting is the case in which only unlabeled data

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \tag{6}$$

are available for training a model. This setting is arguably the most common setting in AD [159], [161], [165], [168]. We will usually assume that the data points have been drawn in an i.i.d. fashion from the data-generating distribution $\mathbb{P}$. For simplicity, we have so far assumed that the data-generating distribution is the same as the normal data distribution $\mathbb{P} \equiv \mathbb{P}^+$. This is often expressed by the statement that the training data is "clean." In practice, however, the data-generating distribution $\mathbb{P}$ may contain noise or contamination.

Noise, in the classical sense, is some inherent source of randomness ε that is added to the signal in the data-generating process, that is, samples from $\mathbb{P}$ have the form $\boldsymbol{x} + \varepsilon$, where $\boldsymbol{x} \sim \mathbb{P}^+$. Noise might be present due to irreducible measurement uncertainties in an application, for example. The greater the noise, the harder it is to accurately estimate the ground-truth level sets of $\mathbb{P}^+$ since informative normal features get obfuscated [165]. This is because added noise expands the regions covered by the observed data in input space $\mathcal{X}$. A standard assumption about noise is that it is unbiased ($\mathbb{E}[\varepsilon] = 0$) and spherically symmetric.

In addition to noise, the contamination (or pollution) of the unlabeled data with undetected anomalies is another important source of the disturbance. For instance, some unnoticed anomalous degradation in an industrial machine might have already occurred during the data collection process. In this case, the data-generating distribution $\mathbb{P}$ is a mixture of the normal data and the anomaly distribution, that is, $\mathbb{P} \equiv (1-\eta)\,\mathbb{P}^+ + \eta\,\mathbb{P}^-$ with contamination (or pollution) rate $\eta \in (0,1)$. The greater the contamination, the more the normal data decision boundary will be distorted by including the anomalous points.

In summary, a more general and realistic assumption is that samples from the data-generating distribution $\mathbb{P}$ have the form of $\boldsymbol{x} + \varepsilon$, where $\boldsymbol{x} \sim (1-\eta)\,\mathbb{P}^+ + \eta\,\mathbb{P}^-$ and ε is the random noise. Assumptions on both the noise distribution ε and contamination rate η are crucial for modeling a specific AD problem. Robust methods [5], [127], [227] specifically aim to account for these sources of disturbance. Note also that, by increasing the level α in the density level set definition above, a corresponding model generally becomes more robust (often at the cost of a higher false alarm rate) since the target decision boundary becomes tighter and excludes the contamination.

3) Semisupervised Setting: The SSAD setting is the case in which both unlabeled and labeled data

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \text{ and } (\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \ldots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y} \tag{7}$$

are available for training a model with $\mathcal{Y} = \{\pm 1\}$, where we denote $\tilde{y} = +1$ for normal and $\tilde{y} = -1$ for anomalous points, respectively. Usually, we have $m \ll n$ in the semisupervised setting, that is, most of the data are unlabeled and only a few labeled instances are available, since labels are often costly to obtain in terms of resources (time, money, and so on). Labeling might, for instance, require domain experts, such as medical professionals (e.g., pathologists) or technical experts (e.g., aerospace engineers). Anomalous instances, in particular, are also infrequent by nature (e.g., rare medical conditions) or very costly (e.g., the failure of some industrial machine). The deliberate generation of anomalies is mostly not an option. However, including known anomalous examples, if available, can significantly improve the detection performance of a model [144], [224], [228]–[231]. Labels are also, sometimes, available in the online setting where alarms raised by the anomaly detector have been investigated to

[2] Strictly speaking, we are assuming that there always exists some data-enclosing hypercube of numerically meaningful values such that the data space $\mathcal{X}$ is bounded and the uniform distribution is well-defined.

determine whether they were correct. Some unsupervised AD methods can be incrementally updated when such labels become available [232]. A recent approach called Outlier Exposure (OE) [233] follows the idea of using large quantities of unlabeled data that are available in some domains as auxiliary anomalies (e.g., online stock photos for computer vision or the English Wikipedia for NLP), thereby effectively labeling this data with $\tilde{y} = -1$. In this setting, we frequently have that $m \gg n$, but this labeled data have increased uncertainty in the labels as the auxiliary data may not only contain anomalies and may not be representative of test time anomalies. We will discuss this specific setting in Sections IV-E and IX-E in more detail. Verifying unlabeled samples as indeed being normal can often be easier due to the more frequent nature of normal data. This is one of the reasons why the special semisupervised case of LPUE [234]–[236], that is, labeled normal and unlabeled examples, is also studied specifically in the AD literature [148], [161], [237]–[239].

Previous work [161] has also referred to the special case of learning exclusively from positive examples as the "SSAD" setting, which is confusing terminology. Although meticulously curated normal data can, sometimes, be available (e.g., in open-category detection [240]), such a setting in which entirely (and confidently) labeled normal examples are available is rather rare in practice. The analysis of this setting is rather again justified by the assumption that most of the given (unlabeled) training data are normal but not the absolute certainty thereof. This makes this setting effectively equivalent to the unsupervised setting from a modeling perspective, apart from maybe weakened assumptions on the level of noise or contamination, which previous works also point out [161]. We, therefore, refer to the more general setting as presented in (7) as the SSAD setting, which incorporates both labeled normal and anomalous examples in addition to unlabeled instances, since this setting is reasonably common in practice. If some labeled anomalies are available, the modeling assumptions about the anomalous distribution $\mathbb{P}^-$, as mentioned in Section II-C1, become critical for effectively incorporating anomalies into training. These include, for instance, whether modes or clusters are expected among the anomalies (e.g., group anomalies).

4) Supervised Setting: The supervised AD setting is the case in which completely labeled data

$$(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y} \tag{8}$$

are available for training a model, where, again, $\mathcal{Y} = \{\pm 1\}$ with $\tilde{y} = +1$ denoting normal instances and $\tilde{y} = -1$ denoting anomalies, respectively. If both the normal and anomalous data points are assumed to be representative for the normal data distribution $\mathbb{P}^+$ and anomaly distribution $\mathbb{P}^-$, respectively, this learning problem is equivalent to supervised binary classification. Such a setting would, thus, not be an AD problem in the strict sense but rather a classification task. Although anomalous modes or clusters might exist, that is, some anomalies might be more likely to occur than others, anything not normal is, by definition, an anomaly. Labeled anomalies are therefore rarely fully representative of some "anomaly class." This distinction is also reflected in modeling: in classification, the objective is to learn a (well-generalizing) decision boundary that best separates the data according to some (closed set of) class labels, but the objective in AD remains the estimation of the normal density level set boundaries. Hence, we should interpret supervised AD problems as label-informed density level set estimation in which confident normal (in-distribution) and anomalous out-of-distribution (OOD) training examples are available. Due to the above and also the high costs often involved with labeling, the supervised AD setting is the most uncommon setting in practice.

Finally, we note that labels may also carry more granular information beyond simply indicating whether some point $\tilde{\boldsymbol{x}}$ is normal ($\tilde{y} = +1$) or anomalous ($\tilde{y} = -1$). In OOD detection [241] or open-category detection [240] problems, for example, the goal is to train a classifier while also detecting examples that are not from any of the known training set classes. In these problems, the labeled data $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m)$ with $\tilde{y} \in \{1, \dots, k\}$ also hold information about the k (sub)classes of the in-distribution $\mathbb{P}^+$. Such information about the structure of the normal data distribution has been shown to be beneficial for semantic detection tasks [242], [243]. We will discuss such specific and related detection problems later in Section IX-B.

5) Further Data Properties: Besides the settings described above, the intrinsic properties of the data itself are also crucial for modeling a specific AD problem. We give a list of relevant data properties in Table 1 and present a toy data set with a specific realization of these properties in Fig. 4, which will serve us as a running example. The assumptions about these properties should be reflected in the modeling choices, such as adding context or deciding among suitable deep or shallow feature maps, which can be challenging. We outline these and further challenges in AD in the following.

D. Challenges in Anomaly Detection

We conclude our introduction by briefly highlighting some notable challenges in AD, some of which directly arise from the definition and data characteristics detailed above. Certainly, the fundamental challenge in AD is the mostly unsupervised nature of the problem, which necessarily requires assumptions to be made about the specific application, the domain, and the given data. These include assumptions about the relevant types of anomalies (see Section II-B2), possible prior assumptions about the anomaly distribution (see Section II-C1) and, if available, the challenge of how to incorporate labeled data instances in a generalizing way (see Sections II-C3 and II-C4).

Table 1 Data Properties Relevant in AD

Data Property	Description
Size $n + m$	Is algorithm scalability in dataset size critical? Are there labeled samples ($m > 0$) for (semi-)supervision?
Dimension D	Low- or high-dimensional? Truly high-dimensional or embedded in some higher dimensional ambient space?
Type	Continuous, discrete, or categorical?
Scales	Are features uni- or multi-scale?
Modality	Uni- or multimodal (classes and clusters)? Is there a hierarchy of sub- and superclasses (or -clusters)?
Convexity	Is the data support convex or non-convex?
Correlation	Are features (linearly or non-linearly) correlated?
Manifold	Has the data a (linear, locally linear, or non-linear) subspace or manifold structure? Are there invariances (translation, rotation, etc.)?
Hierarchy	Is there a natural feature hierarchy (e.g., images, video, text, speech, etc.)? Are low-level or high-level (semantic) anomalies relevant?
Context	Are there contextual features (e.g., time, space, sequence, graph, etc.)? Can anomalies be contextual?
Stationarity	Is the distribution stationary or non-stationary? Is a domain or covariate shift expected?
Noise	Is the noise level ε large or small? Is the noise type Gaussian or more complex?
Contamination	Is the data contaminated with anomalies? What is the contamination rate η?

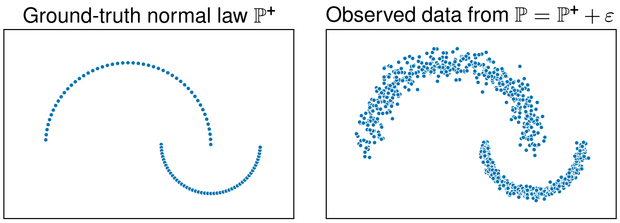


Fig. 4. ***Two-dimensional Big Moon, Small Moon toy example with real-valued ground-truth normal law $\mathbb{P}^+$ that is composed of two 1-D manifolds (bimodal, two-scale, and nonconvex). The unlabeled training data ($n = 1000$ and $m = 0$) are generated from $\mathbb{P} = \mathbb{P}^+ + \varepsilon$, which is subject to Gaussian noise ε. These toy data are nonhierarchical, context-free, and stationary. Anomalies are off-manifold points that may occur uniformly over the displayed range.***

Further questions include how to derive an anomaly score or threshold in a specific task (see Section II-B7)? What level α (see Section II-B8) strikes a balance between false alarms and missed anomalies that is reasonable for the task? Is the data-generating process subject to noise or contamination (see Section II-C2), that is, is robustness a critical aspect? Moreover, identifying and including the data properties given in Table 1 into a method and model can pose challenges as well. The computational complexity in both the data set size $n + m$ and dimensionality D, as well as the memory cost of a model at training time, but also at test time, can be a limiting factor (e.g., for data streams or in real-time monitoring [244]). Is the data-generating process assumed to be nonstationary [245]–[247] and are there distributional shifts expected at test time? For (truly) high-dimensional data, the curse of dimensionality and the resulting concentration of distances can be a major issue [165]. Here, finding a representation that captures the features that are relevant for the task and meaningful for the data and domain becomes vital. Deep AD methods further entail new challenges, such as an increased number of hyperparameters and the selection of suitable network architecture and optimization parameters (learning rate, batch sizes, and so on). In addition, the more complex the data or a model is, the greater the challenges of model interpretability (e.g., [248]–[251]) and decision transparency become. We illustrate some of these practical challenges and provide guidelines with worked-through examples in Section VIII.

Considering the various facets of the AD problem that we have covered in this introduction, it is not surprising that there is a wealth of literature and approaches on the topic. We outline these approaches in the following, where we first examine density estimation and probabilistic models (see Section III), followed by one-class classification methods (see Section IV), and, finally, reconstruction models (see Section V). In these sections, we will point out the connections between deep and shallow methods. Fig. 5 gives an overview and intuition of the approaches. Afterward, in Section VI, we present our unifying view, which will enable us to systematically identify open challenges and paths for future research.

III. DENSITY ESTIMATION AND PROBABILISTIC MODELS

The first category of methods that we introduce predicts anomalies through estimation of the normal data probability distribution. The wealth of existing probability models is, therefore, a clear candidate for the task of AD. This includes classic density estimation methods [252] and deep statistical models. In the following, we describe the adaptation of these techniques to AD.

A. Classic Density Estimation

One of the most basic approaches to multivariate AD is to compute the Mahalanobis distance from a test point to the training data mean [253]. This is equivalent to fitting a multivariate Gaussian distribution to the training data and evaluating the log-likelihood of a test point according to that model [254]. Compared to modeling each dimension of the data independently, fitting a multivariate Gaussian captures linear interactions between pairs of dimensions. To model more complex distributions, nonparametric density estimators have been introduced, such as KDE [12], [252], histogram estimators, and GMMs [255], [256]. The KDE is arguably the most widely used nonparametric density estimator due to theoretical advantages over histograms [257] and the practical issues with fitting and parameter selection for GMMs [258]. The standard KDE, along with a more recent adaptation that can deal with modest levels of outliers in the training data [259], [260], is, therefore, a popular approach to AD. A GMM with

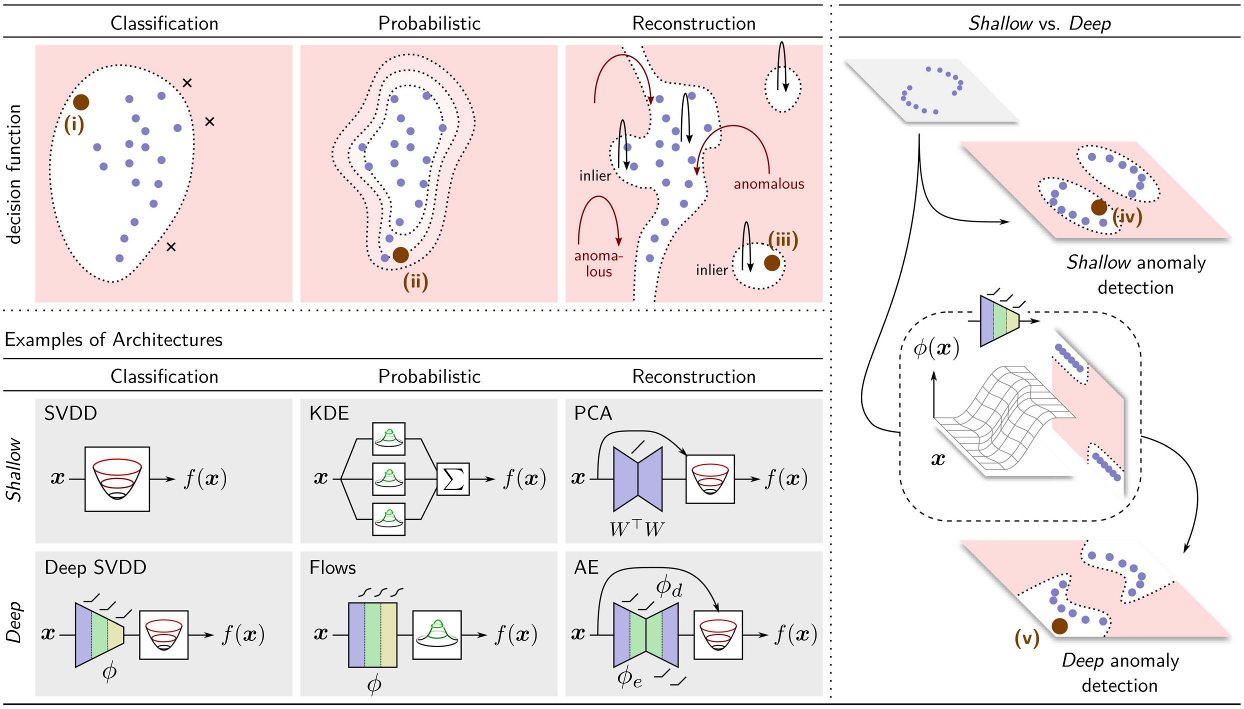


Fig. 5. *Overview of the different approaches to AD. Top: typical decision functions learned by the different AD approaches, where white corresponds to normal and red to anomalous decision regions. One-class classification models typically learn a discriminative decision boundary, probabilistic models a density, and reconstruction models some underlying geometric structure of the data (e.g., manifold or prototypes). Right: deep feature maps enable to learn more flexible, nonlinear decision functions suitable for more complex data. Bottom: diagrams of architectures for a selection of different methods with deep and shallow feature maps. Points (i)–(v): locations in input space, where we highlight some model-specific phenomena. (i) Too loose, the biased one-class boundary may leave anomalies undetected. (ii) Probabilistic models may underfit (or overfit) the tails of a distribution. (iii) Manifold or prototype structure artifacts may result in a good reconstruction of anomalies. (iv) Simple shallow models may fail to fit complex, nonlinear distributions. (v) Compression artifacts of deep feature maps may create "blind spots" in input space.*

a finite number of K mixtures can also be viewed as a soft (probabilistic) clustering method that assumes K prototypical modes (see Section V-A2). This has been used, for example, to represent typical states of a machine in predictive maintenance [261].

While classic nonparametric density estimators perform fairly well for low-dimensional problems, they suffer notoriously from the curse of dimensionality: the sample size required to attain a fixed level of accuracy grows exponentially in the dimension of the feature space. One goal of deep statistical models is to overcome this challenge.

B. Energy-Based Models

Some of the earliest deep statistical models are EBMs [262]–[264]. An EBM is a model whose density is characterized by an energy function $E_\theta(\boldsymbol{x})$ with

$$p_\theta(\boldsymbol{x}) = \frac{1}{Z(\theta)} \exp(-E_\theta(\boldsymbol{x})) \tag{9}$$

where $Z(\theta) = \int \exp(-E_\theta(\boldsymbol{x}))\, d\boldsymbol{x}$ is the so-called partition function that ensures that p_θ integrates to 1. These models are typically trained via gradient descent, approximating the log-likelihood gradient $\nabla_\theta \log p_\theta(\boldsymbol{x})$ via MCMC [265] or SGLD [266], [267]. While one typically cannot evaluate the density p_θ directly due to the intractability of the partition function $Z(\theta)$, the function E_θ can be used as an anomaly score since it is monotonically decreasing as the density p_θ increases.

Early deep EBMs, such as deep belief networks [268] and deep Boltzmann machines [269], are graphical models consisting of layers of latent states followed by an observed output layer that models the training data. Here, the energy function depends not only on the input $\boldsymbol{x}$, but also on a latent state $\boldsymbol{z}$, so the energy function has the form $E_\theta(\boldsymbol{x}, \boldsymbol{z})$. While including latent states allows these approaches to richly model latent probabilistic dependencies in data distributions, these approaches are not particularly amenable to AD since one must marginalize out the latent variables to recover some value related to the likelihood. Later studies replaced the probabilistic latent layers with deterministic ones [270] allowing for the practical evaluation of $E_\theta(\boldsymbol{x})$ for use as an anomaly score. This sort of model has been successfully used for deep AD [271]. Recently, EBMs have also been suggested as a framework to reinterpret deep classifiers where the energy-based training has shown to improve robustness and OOD detection performance [267].

C. Neural Generative Models (VAEs and GANs)

Neural generative models aim to learn a neural network that maps vectors sampled from a simple predefined source distribution $\mathbb{Q}$, usually a Gaussian or uniform distribution, to the actual input distribution $\mathbb{P}^+$. More formally, the objective is to train the network so that $\phi_\omega(\mathbb{Q}) \approx \mathbb{P}^+$, where $\phi_\omega(\mathbb{Q})$ is the distribution that results from pushing the source distribution $\mathbb{Q}$ through neural network ϕ_ω. The two most established neural generative models are VAEs [272]–[274] and GANs [275].

1) VAEs: VAEs learn deep latent-variable models where the inputs $\boldsymbol{x}$ are parameterized on latent samples $\boldsymbol{z} \sim \mathbb{Q}$ via some neural network, so as to learn a distribution $p_\theta(\boldsymbol{x} \,|\, \boldsymbol{z})$ such that $p_\theta(\boldsymbol{x}) \approx p^+(\boldsymbol{x})$. A common instantiation of this is to let $\mathbb{Q}$ be an isotropic multivariate Gaussian distribution and let the neural network $\phi_{d,\omega} = (\boldsymbol{\mu}_\omega, \boldsymbol{\sigma}_\omega)$ (the decoder) with weights ω parameterize the mean and variance of an isotropic Gaussian distribution, so $p_\theta(\boldsymbol{x} \,|\, \boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_\omega(\boldsymbol{z}), \boldsymbol{\sigma}^2_\omega(\boldsymbol{z})I)$. Performing maximum likelihood estimation on θ is typically intractable. To remedy this, an additional network $\phi_{e,\omega'}$ (the encoder) is introduced to parameterize a variational distribution $q_{\theta'}(\boldsymbol{z} \,|\, \boldsymbol{x})$, with θ' encapsulated by the output of $\phi_{e,\omega'}$, to approximate the latent posterior $p(\boldsymbol{z} \,|\, \boldsymbol{x})$. The full model is then optimized via the ELBO in a variational Bayes manner

$$\max_{\theta,\theta'} \; -D_{\mathrm{KL}}(q_{\theta'}(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})) + \mathbb{E}_{q_{\theta'}(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]. \quad (10)$$

Optimization proceeds using stochastic gradient variational Bayes [272]. Given a trained VAE, one can estimate $p_\theta(\boldsymbol{x})$ via Monte Carlo sampling from the prior $p(\boldsymbol{z})$ and computing $\mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}[p_\theta(\boldsymbol{x} \,|\, \boldsymbol{z})]$. Using this score directly for AD has a nice theoretical interpretation, but experiments have shown that it tends to perform worse [276], [277] than alternatively using the reconstruction probability [278], which conditions on $\boldsymbol{x}$ to estimate $\mathbb{E}_{q_{\theta'}(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$. The latter can also be seen as a probabilistic reconstruction model using a stochastic encoding and decoding process (see Section V-C).

2) GANs: GANs pose the problem of learning the target distribution as a zero-sum-game: a generative model is trained in competition with an adversary that challenges it to generate samples whose distribution is similar to the training distribution. A GAN consists of two neural networks, a generator network $\phi_\omega : \mathcal{Z} \to \mathcal{X}$, and a discriminator network $\psi_{\omega'} : \mathcal{X} \to (0,1)$ that are pitted against each other so that the discriminator is trained to discriminate between $\phi_\omega(\boldsymbol{z})$ and $\boldsymbol{x} \sim \mathbb{P}^+$, where $\boldsymbol{z} \sim \mathbb{Q}$. The generator is trained to fool the discriminator network, thereby encouraging the generator to produce samples more similar to the target distribution. This is done using the following adversarial objective:

$$\begin{aligned} \min_{\omega} \max_{\omega'} \; & \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}^+}[\log \psi_{\omega'}(\boldsymbol{x})] \\ & + \mathbb{E}_{\boldsymbol{z} \sim \mathbb{Q}}[\log(1 - \psi_{\omega'}(\phi_\omega(\boldsymbol{z})))]. \end{aligned} \quad (11)$$

Training is typically carried out via an alternating optimization scheme, which is notoriously finicky [279]. There exist many GAN variants, for example, the Wasserstein GAN [280], [281], which is frequently used for AD methods using GANs, and StyleGAN, which has produced impressive high-resolution photorealistic images [92].

Due to their construction, GAN models offer no way to assign a likelihood to points in the input space. Using the discriminator directly has been suggested as one approach to use GANs for AD [138], which is conceptually close to one-class classification (see Section IV). Other approaches apply optimization to find a point $\tilde{\boldsymbol{z}}$ in latent space $\mathcal{Z}$ such that $\tilde{\boldsymbol{x}} \approx \phi_\omega(\tilde{\boldsymbol{z}})$ for the test point $\tilde{\boldsymbol{x}}$. The authors of AnoGAN [50] recommend using an intermediate layer of the discriminator, $f_{\omega'}$, and setting the anomaly score to be a convex combination of the reconstruction loss $\|\tilde{\boldsymbol{x}} - \phi_\omega(\tilde{\boldsymbol{z}})\|$ and the discrimination loss $\|f_{\omega'}(\tilde{\boldsymbol{x}}) - f_{\omega'}(\phi_\omega(\tilde{\boldsymbol{z}}))\|$. In AD-GAN [147], the authors recommend initializing the search for latent points multiple times to find a collection of m latent points $\tilde{\boldsymbol{z}}_1, \ldots, \tilde{\boldsymbol{z}}_m$ while simultaneously adapting the network parameters ω_i individually for each $\tilde{\boldsymbol{z}}_i$ to improve the reconstruction and using the mean reconstruction loss as an anomaly score

$$\frac{1}{m} \sum_{i=1}^{m} \|\tilde{\boldsymbol{x}} - \phi_{\omega_i}(\tilde{\boldsymbol{z}}_i)\|. \quad (12)$$

Viewing the generator as a stochastic decoder and the search for an optimal latent point $\tilde{\boldsymbol{z}}$ as an (implicit) encoding of a test point $\tilde{\boldsymbol{x}}$, utilizing a GAN this way with the reconstruction error for AD is similar to reconstruction methods, particularly AEs (see Section V-C). Later GAN adaptations have added explicit encoding networks that are trained to find the latent point $\tilde{\boldsymbol{z}}$. This has been used in a variety of ways, usually again incorporating the reconstruction error [56], [148], [151].

D. Normalizing Flows

Like neural generative models, normalizing flows [282]–[284] attempt to map data points from a source distribution $\boldsymbol{z} \sim \mathbb{Q}$ (usually called base distribution for normalizing flows) so that $\boldsymbol{x} \approx \phi_\omega(\boldsymbol{z})$ is distributed according to p^+. The crucial distinguishing characteristic of normalizing flows is that the latent samples are D-dimensional, so they have the same dimensionality as the input space, and the network consists of L layers $\phi_{i,\omega_i} : \mathbb{R}^D \to \mathbb{R}^D$, so $\phi_\omega = \phi_{L,\omega_L} \circ \cdots \circ \phi_{1,\omega_1}$, where each ϕ_{i,ω_i} is designed to be invertible for all ω_i, thereby making the entire network invertible. The benefit of this formulation is that

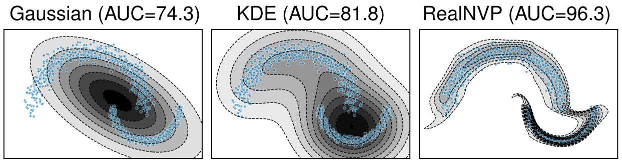


Fig. 6. ***Density estimation models on the Big Moon, Small Moon toy example (see Fig. 4). The parametric Gaussian model is limited to an ellipsoidal (convex, unimodal) density. KDE with an RBF kernel is more flexible, yet tends to underfit the (multiscale) distribution due to a uniform kernel scale. RealNVP is the most flexible model, yet flow architectures induce biases as well, here a connected support caused by affine coupling layers in RealNVP.***

the probability density of $\boldsymbol{x}$ can be calculated exactly via a change of variables

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{z}}(\phi_\omega^{-1}(\boldsymbol{x})) \prod_{i=1}^{L} \left| \det J\phi_{i,\omega_i}^{-1}(\boldsymbol{x}_i) \right| \tag{13}$$

where $\boldsymbol{x}_L = \boldsymbol{x}$ and $\boldsymbol{x}_i = \phi_{i+1}^{-1} \circ \cdots \circ \phi_L^{-1}(\boldsymbol{x})$ otherwise. Normalizing flow models are typically optimized to maximize the likelihood of the training data. Evaluating each layer's Jacobian and its determinant can be very expensive. Consequently, the layers of flow models are usually designed so that the Jacobian is guaranteed to be upper (or lower) triangular or have some other nice structure such that one does not need to compute the full Jacobian to evaluate its determinant [282], [285], [286] (see [287] for an application in physics).

An advantage of these models over other methods is that one can calculate the likelihood of a point directly without any approximation while also being able to sample from it reasonably efficiently. Because the density $p_{\boldsymbol{x}}(\boldsymbol{x})$ can be computed exactly, normalizing flow models can be applied directly for AD [288], [289].

A drawback of these models is that they do not perform any dimensionality reduction, which argues against applying them to images where the true (effective) dimensionality is much smaller than the image dimensionality. Furthermore, it has been observed that these models often assign a high likelihood to anomalous instances [277]. Recent work suggests that one reason for this seems to be that the likelihood in current flow models is dominated by low-level features due to specific network architecture inductive biases [243], [290]. Despite present limitations, we have included normalizing flows here because we believe that they may provide an elegant and promising direction for future AD methods. We will come back to this in our outlook in Section IX.

E. Discussion

Above, we have focused on the case of density estimation on i.i.d. samples of low-dimensional data and images. For comparison, we show in Fig. 6 three canonical density estimation models (Gaussian, KDE, and RealNVP) trained on the *Big Moon, Small Moon* toy data set, each of which makes use of a different feature representation (raw input, kernel, and neural network). It is worth noting that there exist many deep statistical models for other settings. When performing conditional AD, for example, one can use GAN [291], VAE [292], and normalizing flow [293] variants that perform conditional density estimation. Likewise, there exist many DGMs for virtually all data types, including time-series data [292], [294], text [295], [296], and graphs [297]–[299], all of which may potentially be used for AD.

It has been argued that full density estimation is not needed for solving the AD problem since one learns all density level sets simultaneously when one really only needs a single density level set [6], [7], [216]. This violates Vapnik's principle: "[W]hen limited amount of data is available, one should avoid solving a more general problem as an intermediate step to solve the original problem" [300]. The methods in Section IV seek to compute only a single density level set, that is, they perform one-class classification.

IV. ONE-CLASS CLASSIFICATION

One-class classification [223], [224], [301]–[303], occasionally also called single-class classification [304], [305], adopts a discriminative approach to AD. Methods based on one-class classification try to avoid a full estimation of the density as an intermediate step to AD. Instead, these methods aim to directly learn a decision boundary that corresponds to a desired density level set of the normal data distribution $\mathbb{P}^+$, or more generally, to produce a decision boundary that yields a low error when applied to unseen data.

A. One-Class Classification Objective

We can see one-class classification as a particularly tricky classification problem, namely as binary classification where we only have (or almost only have) access to data from one class—the normal class. Given this imbalanced setting, the one-class classification objective is to learn a one-class decision boundary that minimizes: 1) falsely raised alarms for true normal instances (i.e., the false alarm rate or type I error) and 2) undetected or missed true anomalies (i.e., the miss rate or type II error). Achieving a low (or zero) false alarm rate is conceptually simple: given enough normal data points, one could just draw some boundary that encloses all the points, for example, a sufficiently large ball that contains all data instances. The crux here is, of course, to simultaneously keep the miss rate low, that is, to not draw this boundary too loosely. For this reason, one usually *a priori* specifies some target false alarm rate $\alpha \in [0, 1]$ for which the miss rate is then sought to be minimized. Note that this precisely corresponds to the idea of estimating an α-density level set for some *a priori* fixed level $\alpha \in [0, 1]$. The key question in one-class classification, thus, is how to minimize the miss rate for some given target false alarm rate with access to no (or only a few) anomalies.

We can express the rationale above in terms of the binary classification risk [212], [222]. Let $Y \in \{\pm 1\}$ be the class random variable, where again $Y = +1$ denotes normal and $Y = -1$ denotes anomalous points, so we can then identify the normal data distribution as $\mathbb{P}^{+} \equiv \mathbb{P}_{X|Y=+1}$ and the anomaly distribution as $\mathbb{P}^{-} \equiv \mathbb{P}_{X|Y=-1}$, respectively. Furthermore, let $\ell : \mathbb{R} \times \{\pm 1\} \to \mathbb{R}$ be a binary classification loss and $f : \mathcal{X} \to \mathbb{R}$ be some real-valued score function. The classification risk of f under loss ℓ is then given by

$$R(f) = \mathbb{E}_{X\sim\mathbb{P}^{+}}[\ell(f(X), +1)] + \mathbb{E}_{X\sim\mathbb{P}^{-}}[\ell(f(X), -1)]. \quad (14)$$

Minimizing the second term—the expected loss of classifying true anomalies as normal—corresponds to minimizing the (expected) miss rate. Given some unlabeled data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$ and, potentially, some additional labeled data $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \dots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m)$, we can apply the principle of empirical risk minimization to obtain

$$\min_{f} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), +1) + \frac{1}{m}\sum_{j=1}^{m} \ell(f(\tilde{\boldsymbol{x}}_j), \tilde{y}_j) + \mathcal{R}. \quad (15)$$

This solidifies the empirical one-class classification objective. Note that the second term is an empty sum in the unsupervised setting. Without any additional constraints or regularization, the empirical objective (15) would then be trivial. We add $\mathcal{R}$ as an additional term to denote and capture regularization, which may take various forms depending on the assumptions about f but critically also about $\mathbb{P}^{-}$. Generally, the regularization $\mathcal{R} = \mathcal{R}(f)$ aims to minimize the miss rate (e.g., via volume minimization and assumptions about $\mathbb{P}^{-}$) and improve generalization (e.g., via smoothing of f). Furthermore, note that the pseudolabeling of $y = +1$ in the first term incorporates the assumption that the n unlabeled training data points are normal. This assumption can be adjusted, however, through specific choices of the loss (e.g., hinge) and regularization, for example, requiring some fraction of the unlabeled data to get misclassified to include an assumption about the contamination rate η or achieve some target false alarm rate α.

B. One-Class Classification in Input Space

As an illustrative example that conveys useful intuition, consider the simple idea from above of fitting a data-enclosing ball as a one-class model. Given $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, we can define the following objective:

$$\begin{aligned} &\min_{R,\boldsymbol{c},\boldsymbol{\xi}} \quad R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i \\ &\text{s.t. } \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \le R^2 + \xi_i, \quad \xi_i \ge 0 \quad \forall i. \end{aligned} \quad (16)$$

In other words, we aim to find a hypersphere with radius $R > 0$ and center $\boldsymbol{c} \in \mathcal{X}$ that encloses the data ($\|\boldsymbol{x}_i - \boldsymbol{c}\|^2 \le R^2$). To control the miss rate, we minimize the volume of this hypersphere by minimizing R^2 to achieve a tight spherical boundary. Slack variables $\xi_i \ge 0$ allow some points to fall outside the sphere, thus making the boundary soft, where hyperparameter $\nu \in (0, 1]$ balances this tradeoff.

Objective (16) exactly corresponds to SVDD applied in the input space $\mathcal{X}$, motivated above as in [7], [223], and [224]. Equivalently, we can derive (16) from the binary classification risk. Consider the (shifted, cost-weighted) hinge loss $\ell(s, y)$ defined by $\ell(s, +1) = (1/(1 + \nu)) \max(0, s)$ and $\ell(s, -1) = (\nu/(1 + \nu)) \max(0, -s)$ [222]. Then, for a hypersphere model $f_\theta(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{c}\|^2 - R^2$ with parameters $\theta = (R, \boldsymbol{c})$, the corresponding classification risk (14) is given by

$$\begin{aligned} \min_{\theta}\ &\mathbb{E}_{X\sim\mathbb{P}^{+}}[\max(0, \|X - \boldsymbol{c}\|^2 - R^2)] \\ &+\nu\, \mathbb{E}_{X\sim\mathbb{P}^{-}}[\max(0, R^2 - \|X - \boldsymbol{c}\|^2)]. \end{aligned} \quad (17)$$

We can estimate the first term in (17) empirically from $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, again assuming that (most of) these points have been drawn from $\mathbb{P}^{+}$. If labeled anomalies are absent, we can still make an assumption about their distribution $\mathbb{P}^{-}$. Following the basic, uninformed prior assumption that anomalies may occur uniformly on $\mathcal{X}$ (i.e., $\mathbb{P}^{-} \equiv \mathcal{U}(\mathcal{X})$), we can examine the expected value in the second term analytically:

$$\begin{aligned} &\mathbb{E}_{X\sim\mathcal{U}(\mathcal{X})}[\max(0, R^2 - \|X - \boldsymbol{c}\|^2)] \\ &\quad = \frac{1}{\lambda(\mathcal{X})}\int_{\mathcal{X}} \max(0, R^2 - \|\boldsymbol{x} - \boldsymbol{c}\|^2)\, \mathrm{d}\lambda(\boldsymbol{x}) \\ &\quad \le R^2\, \frac{\lambda(\mathcal{B}_R(\boldsymbol{c}))}{\lambda(\mathcal{X})} \le R^2 \end{aligned} \quad (18)$$

where $\mathcal{B}_R(\boldsymbol{c}) \subseteq \mathcal{X}$ denotes the ball centered at $\boldsymbol{c}$ with radius R and λ is again the standard (Lebesgue) measure of volume.[3] This shows that the minimum volume principle [218], [220] naturally arises in one-class classification through seeking to minimize the risk of missing anomalies, here illustrated for an assumption that the anomaly distribution $\mathbb{P}^{-}$ follows a uniform distribution. Overall, from (17), we, thus, can derive the empirical objective

$$\min_{R,\boldsymbol{c}}\ R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \max(0, \|\boldsymbol{x}_i - \boldsymbol{c}\|^2 - R^2) \quad (19)$$

which corresponds to (16) with the constraints directly incorporated into the objective function. We remark that the cost-weighting hyperparameter $\nu \in (0, 1]$ is

[3] Again note that we assume $\lambda(\mathcal{X}) < \infty$ here, that is, the data space $\mathcal{X}$ can be bounded to numerically meaningful values.

purposefully chosen here since it is an upper bound on the ratio of points outside and a lower bound on the ratio of points inside or on the boundary of the sphere [6], [137]. We can, therefore, see ν as an approximation of the false alarm rate, that is, $\nu \approx \alpha$.

A sphere in the input space $\mathcal{X}$ is, of course, a very limited model and only matches a limited class of distributions $\mathbb{P}^+$ (e.g., an isotropic Gaussian distribution). MVEs [178], [306] and the MCD estimator [307] are a generalization to nonisotropic distributions with elliptical support. Nonparametric methods, such as one-class neighbor machines [308], provide additional freedom to model multimodal distributions having nonconvex support. Extending the objective and principles above to general feature spaces (e.g., [211], [300], and [309]) further increases the flexibility of one-class models and enables decision boundaries for more complex distributions.

C. Kernel-Based One-Class Classification

The kernel-based OC-SVM [6], [310] and SVDD [7], [224] are perhaps the most well-known one-class classification methods. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be some PSD kernel with associated RKHS $\mathcal{F}_k$ and corresponding feature map $\phi_k : \mathcal{X} \to \mathcal{F}_k$, so $k(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \langle \phi_k(\boldsymbol{x}), \phi_k(\tilde{\boldsymbol{x}}) \rangle$ for all $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{X}$. The objective of (kernel) SVDD is again to find a data-enclosing hypersphere of minimum volume. The SVDD primal problem is the one given in (16) but with the hypersphere model $f_\theta(\boldsymbol{x}) = \|\phi_k(\boldsymbol{x}) - \mathbf{c}\|^2 - R^2$ defined in feature space $\mathcal{F}_k$ instead. In comparison, the OC-SVM objective is to find a hyperplane $\boldsymbol{w} \in \mathcal{F}_k$ that separates the data in feature space $\mathcal{F}_k$ with maximum margin from the origin

$$\begin{aligned} &\min_{\boldsymbol{w}, \rho, \boldsymbol{\xi}} \; \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i \\ &\text{s.t. } \rho - \langle \phi_k(\boldsymbol{x}_i), \boldsymbol{w} \rangle \leq \xi_i, \quad \xi_i \geq 0 \quad \forall i. \end{aligned} \tag{20}$$

Thus, the OC-SVM uses a linear model $f_\theta(\boldsymbol{x}) = \rho - \langle \phi_k(\boldsymbol{x}), \boldsymbol{w} \rangle$ in feature space $\mathcal{F}_k$ with model parameters $\theta = (\boldsymbol{w}, \rho)$. The margin to the origin is given by $(\rho / \|\boldsymbol{w}\|)$, which is maximized via maximizing ρ, where $\|\boldsymbol{w}\|$ acts as a normalizer.

Both the OC-SVM and SVDD can be solved in their respective dual formulations that are quadratic programs that only involve dot products (the feature map ϕ_k is implicit). For the standard Gaussian kernel (or any kernel with constant norm $k(\boldsymbol{x}, \boldsymbol{x}) = c > 0$), the OC-SVM and SVDD are equivalent [224]. In this case, the corresponding density level set estimator defined by

$$\hat{C}_\nu = \{\boldsymbol{x} \in \mathcal{X} \mid f_\theta(\boldsymbol{x}) < 0\} \tag{21}$$

is, in fact, an asymptotically consistent ν-density level set estimator [311]. The solution paths of hyperparameter ν have been analyzed for both the OC-SVM [312] and SVDD [313].

Kernel-induced feature spaces considerably improve the expressive power of one-class methods and allow learning well-performing models in multimodal, nonconvex, and nonlinear data settings. Many variants of kernel one-class classification have been proposed and studied over the years, such as hierarchical formulations for nested density level set estimation [314], [315], multisphere SVDD [316], multiple kernel learning for OC-SVM [317], [318], OC-SVM for group AD [197], boosting via L_1-norm regularized OC-SVM [319], one-class kernel Fisher discriminants [320]–[322], Bayesian data description [323], and distributed [324], incremental learning [325], or robust [326] variants.

D. Deep One-Class Classification

Selecting kernels and handcrafting relevant features can be challenging and quickly become impractical for complex data. Deep one-class classification methods aim to overcome these challenges by learning useful neural network feature maps $\phi_\omega : \mathcal{X} \to \mathcal{Z}$ from the data or transferring such networks from related tasks. Deep SVDD [137], [144], [145], [327] and deep OC-SVM variants [136], [328] employ a hypersphere model $f_\theta(\boldsymbol{x}) = \|\phi_\omega(\boldsymbol{x}) - \mathbf{c}\|^2 - R^2$ and linear model $f_\theta(\boldsymbol{x}) = \rho - \langle \phi_\omega(\boldsymbol{x}), \boldsymbol{w} \rangle$ with explicit neural feature maps $\phi_\omega(\cdot)$ in (16) and (20), respectively. These methods are typically optimized with SGD variants [329]–[331], which, together with GPU parallelization, makes them scale to large data sets.

The one-class Deep SVDD [137], [332] has been introduced as a simpler variant compared to using a neural hypersphere model in (16), which poses the following objective:

$$\min_{\omega, \boldsymbol{c}} \; \frac{1}{n}\sum_{i=1}^{n} \|\phi_\omega(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 + \mathcal{R}. \tag{22}$$

Here, the neural network transformation $\phi_\omega(\cdot)$ is learned to minimize the mean squared distance over all data points to center $\boldsymbol{c} \in \mathcal{Z}$. Optimizing this simplified objective has been found to converge faster and be effective in many situations [137], [144], [332]. In light of our unifying view, we will see that we may interpret one-class Deep SVDD also as a single-prototype deep clustering method (see Sections V-A2 and V-D).

A recurring question in deep one-class classification is how to meaningfully regularize against a feature map collapse $\phi_\omega \equiv \boldsymbol{c}$. Without regularization, minimum volume or maximum margin objectives, such as (16), (20), or (22), could be trivially solved with a constant mapping [137], [333]. Possible solutions for this include adding a reconstruction term or architectural constraints [137], [327], freezing the embedding [136], [139], [140], [142], [334], inversely penalizing the embedding variance [335], using true [144], [336], auxiliary [139], [233], [332], [337], or artificial [337] negative examples in training,

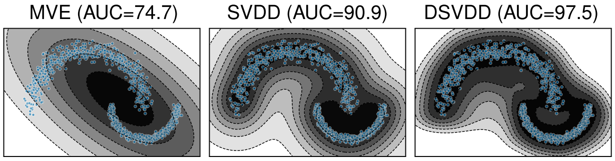


Fig. 7. ***One-class classification models on the Big Moon, Small Moon toy example (see Fig. 4). An MVE in input space is limited to enclose an ellipsoidal, convex region. By (implicitly) fitting a hypersphere in kernel feature space, SVDD enables nonconvex support estimation. Deep SVDD learns an (explicit) neural feature map (here with smooth ELU activations) that extracts multiple data scales to fit a hypersphere model in feature space for support description.***

pseudolabeling [152], [153], [155], [335], or integrating some manifold assumption [333]. Further variants of deep one-class classification include multimodal [145] or time-series extensions [338] and methods that employ adversarial learning [138], [141], [339] or transfer learning [139], [142].

Deep one-class classification methods generally offer greater modeling flexibility and enable the learning or transfer of task-relevant features for complex data. They usually require more data to be effective though or must rely on some informative domain prior (e.g., some pre-trained network). However, the underlying principle of one-class classification methods—targeting a discriminative one-class boundary in learning—remains unaltered, regardless of whether a deep or shallow feature map is used. We show three canonical one-class classification models (MVE, SVDD, and DSVDD) trained on the *Big Moon, Small Moon* toy data set, each using a different feature representation (raw input, kernel, and neural network), in Fig. 7 for comparison.

E. Negative Examples

One-class classifiers can usually incorporate labeled negative examples ($y = -1$) in a direct manner due to their close connection to binary classification, as explained above. Such negative examples can facilitate an empirical estimation of the miss rate [see (14) and (15)]. We here recognize three qualitative types of negative examples that have been studied in the literature, which we distinguish as artificial, auxiliary, and true negative examples that increase in their informativeness in this order.

The idea to approach unsupervised learning problems through generating artificial data points has been around for some time (see [340, Section 14.2.4]). If we assume that the anomaly distribution $\mathbb{P}^-$ has some form that we can generate data from, one idea would be to simply train a binary classifier to discern between the normal and the artificial negative examples. For the uniform prior $\mathbb{P}^- \equiv \mathcal{U}(\mathcal{X})$, this approach yields an asymptotically consistent density level set estimator [212]. However, classification against uniformly drawn points from a hypercube quickly becomes ineffective in higher dimensions. To improve over artificial uniform sampling, more informed sampling strategies have been proposed [341], such as resampling schemes [342], manifold sampling [343], and sampling based on local density estimation [344], [345], as well as active learning strategies [346]–[348]. Another recent idea is to treat the enormous quantities of data that are publicly available in some domains as auxiliary negative examples [233], for example, images from photo-sharing sites for computer vision tasks and the English Wikipedia for NLP tasks. Such auxiliary examples provide more informative domain knowledge, for instance, about the distribution of natural images or the English language, in general, as opposed to sampling random pixels or words. This approach, called OE [233], which trains on known anomalies, can significantly improve deep AD performance in some domains [153], [233]. OE has also been used with density-based methods by employing a margin loss [233] or temperature annealing [243] on the log-likelihood ratio between positive and negative examples. The most informative labeled negative examples are ultimately true anomalies, for example, verified by some domain expert. Access to even a few labeled anomalies has been shown to improve detection performance significantly [144], [224], [229]. There also have been active learning algorithms proposed, which includes subjective user feedback (e.g., from an expert) to learn about the user-specific informativeness of particular anomalies in an application [349]. Finally, we remark that negative examples have also been incorporated heuristically into reconstruction models via using a bounded reconstruction error [350] since maximizing the unbounded error for negative examples can quickly become unstable. We will turn to reconstruction models next.

V. RECONSTRUCTION MODELS

Models that are trained on a reconstruction objective are among the earliest [351], [352] and most common [180], [182] neural network approaches to AD. Reconstruction-based methods learn a model that is optimized to well-reconstruct normal data instances, thereby aiming to detect anomalies by failing to accurately reconstruct them under the learned model. Most of these methods have a purely geometric motivation (e.g., PCA or deterministic AEs), yet some probabilistic variants reveal a connection to density (level set) estimation. In this section, we define the general reconstruction learning objective, highlight common underlying assumptions, present standard reconstruction-based methods, and discuss their variants.

A. Reconstruction Objective

Let $\phi_\theta : \mathcal{X} \to \mathcal{X}, \boldsymbol{x} \mapsto \phi_\theta(\boldsymbol{x})$ be a feature map from the data space $\mathcal{X}$ onto itself that is composed of an encoding function $\phi_e : \mathcal{X} \to \mathcal{Z}$ (the encoder) and a decoding function $\phi_d : \mathcal{Z} \to \mathcal{X}$ (the decoder), that is, $\phi_\theta \equiv (\phi_d \circ \phi_e)_\theta$, where θ holds the parameters of both the encoder and the decoder. We call $\mathcal{Z}$ the latent space and $\phi_e(\boldsymbol{x}) = \boldsymbol{z}$ the latent representation (or embedding or code) of $\boldsymbol{x}$. The reconstruction objective then is to learn ϕ_θ such that $\phi_\theta(\boldsymbol{x}) = \phi_d(\phi_e(\boldsymbol{x})) = \hat{\boldsymbol{x}} \approx \boldsymbol{x}$, that is, to find some encoding and decoding transformation so that $\boldsymbol{x}$ is reconstructed with minimal error, usually measured in the Euclidean distance. Given unlabeled data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, the reconstruction objective is given by

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - (\phi_d \circ \phi_e)_\theta(\boldsymbol{x}_i)\|^2 + \mathcal{R} \tag{23}$$

where $\mathcal{R}$ again denotes the different forms of regularization that various methods introduce, for example, on the parameters θ, the structure of the encoding and decoding transformations, or the geometry of latent space $\mathcal{Z}$. Without any restrictions, the reconstruction objective (23) would be optimally solved by the identity map $\phi_\theta \equiv \mathrm{id}$, but then, of course, nothing would be learned from the data. In order to learn something useful, structural assumptions about the data-generating process are, therefore, necessary. We here identify two principal assumptions: the manifold and the prototype assumptions.

1) Manifold Assumption: The manifold assumption asserts that the data lives (approximately) on some lower dimensional (possibly nonlinear and nonconvex) manifold $\mathcal{M}$ that is embedded within the data space $\mathcal{X}$, that is, $\mathcal{M} \subset \mathcal{X}$ with $\dim(\mathcal{M}) < \dim(\mathcal{X})$. In this case, $\mathcal{X}$ is, sometimes, also called the ambient or observation space. For natural images observed in pixel space, for instance, the manifold captures the structure of scenes, variation due to rotation and translation, and changes in color, shape, size, texture, and so on. For human voices observed in audio-signal space, the manifold captures variation due to the words being spoken and person-to-person variation in the anatomy and physiology of the vocal folds. The (approximate) manifold assumption implies that there exists a lower dimensional latent space $\mathcal{Z}$ and functions $\phi_e : \mathcal{X} \mapsto \mathcal{Z}$ and $\phi_d : \mathcal{Z} \mapsto \mathcal{X}$ such that, for all $x \in \mathcal{X}$, $x \approx \phi_d(\phi_e(x))$. Consequently, the generating distribution $\mathbb{P}$ can be represented as the push-forward through ϕ_d of a latent distribution $\mathbb{P}_Z$. Equivalently, the latent distribution $\mathbb{P}_Z$ is the push-forward of $\mathbb{P}$ through ϕ_e.

The goal of learning is, therefore, to learn the pair of functions ϕ_e and ϕ_d so that $\phi_d(\phi_e(\mathcal{X})) \approx \mathcal{M} \subset \mathcal{X}$. Methods that incorporate the manifold assumption usually restrict the latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ to have much lower dimensionality d than the data space $\mathcal{X} \subseteq \mathbb{R}^D$ (i.e., $d \ll D$). The manifold assumption is also widespread in related unsupervised learning tasks, such as manifold learning itself [353], [354], dimensionality reduction [3], [355]–[357], disentanglement [210], [358], and representation learning, in general [80], [359].

2) Prototype Assumption: The prototype assumption asserts that there exists a finite number of prototypical elements in the data space $\mathcal{X}$ that characterize the data well. We can model this assumption in terms of a data-generating distribution that depends on a discrete latent categorical variable $Z \in \mathcal{Z} = \{1, \ldots, K\}$ that captures some K prototypes or modes of the data distribution. This prototype assumption is also common in clustering and classification when we assume a collection of prototypical instances represent clusters or classes well. With the reconstruction objective under the prototype assumption, we aim to learn an encoding function that, for $\boldsymbol{x} \in \mathcal{X}$, identifies a $\phi_e(\boldsymbol{x}) = k \in \{1, \ldots, K\}$ and a decoding function $k \mapsto \phi_d(k) = \boldsymbol{c}_k$ that maps to some kth prototype (or some prototypical distribution or mixture of prototypes more generally) such that the reconstruction error $\|\boldsymbol{x} - \boldsymbol{c}_k\|$ becomes minimal. In contrast to the manifold assumption where we aim to describe the data by some continuous mapping, under the (most basic) prototype assumption, we characterize the data by a discrete set of vectors $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\} \subseteq \mathcal{X}$. The method of representing a data distribution by a set of prototype vectors is also known as VQ [360], [361].

3) Reconstruction Anomaly Score: A model that is trained on the reconstruction objective must extract salient features and characteristic patterns from the data in its encoding—subject to imposed model assumptions—so that its decoding from the compressed latent representation achieves low reconstruction error (e.g., feature correlations and dependencies, recurring patterns, cluster structure, and statistical redundancy). Assuming that the training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ include mostly normal points, we, therefore, expect a reconstruction-based model to produce a low reconstruction error for normal instances and a high reconstruction error for anomalies. For this reason, the anomaly score is usually also directly defined by the reconstruction error

$$s(\boldsymbol{x}) = \|\boldsymbol{x} - (\phi_d \circ \phi_e)_\theta(\boldsymbol{x})\|^2. \tag{24}$$

For models that have learned some truthful manifold structure or prototypical representation, a high reconstruction error would then detect off-manifold or nonprototypical instances.

Most reconstruction methods do not follow any probabilistic motivation, and a point $\boldsymbol{x}$ gets flagged anomalous simply because it does not conform to its 'idealized' representation $\phi_d(\phi_e(\boldsymbol{x})) = \hat{\boldsymbol{x}}$ under the encoding and decoding processes. However, some reconstruction methods also have probabilistic interpretations, such as PCA [362], or even are derived from probabilistic objectives, such as Bayesian PCA [363] or VAEs [272]. These methods are again related to density (level set) estimation (under specific assumptions about some latent structure), usually in the sense that a high reconstruction error indicates low-density regions, and vice versa.

B. Principal Component Analysis

A common way to formulate the PCA objective is to seek an orthogonal basis W in data space $\mathcal{X} \subseteq \mathbb{R}^D$ that maximizes the empirical variance of the (centered) data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$

$$\max_{W} \sum_{i=1}^{n} \|W\boldsymbol{x}_i\|^2 \quad \text{s.t. } WW^\top = I. \tag{25}$$

Solving this objective results in a well-known eigenvalue problem since the optimal basis is given by the eigenvectors of the empirical covariance matrix where the respective eigenvalues correspond to the componentwise variances [364]. The $d \leq D$ components that explain most of the variance—the principal components—are then given by the d eigenvectors that have the largest eigenvalues.

Several works have adapted PCA for AD [77], [365]–[370], which can be considered the default reconstruction baseline. From a reconstruction perspective, the objective to find an orthogonal projection $W^\top W$ to a d-dimensional linear subspace (which is the case for $W \in \mathbb{R}^{d\times D}$ with $WW^\top = I$) such that the mean squared reconstruction error is minimized

$$\min_{W} \sum_{i=1}^{n} \|\boldsymbol{x}_i - W^\top W\boldsymbol{x}_i\|^2 \quad \text{s.t. } WW^\top = I \tag{26}$$

yields exactly the same PCA solution. Thus, PCA optimally solves the reconstruction objective (23) for a linear encoder $\phi_e(\boldsymbol{x}) = W\boldsymbol{x} = \boldsymbol{z}$ and transposed linear decoder $\phi_d(\boldsymbol{z}) = W^\top\boldsymbol{z}$ with constraint $WW^\top = I$. For linear PCA, we can also readily identify its probabilistic interpretation [362], namely that the data distribution follows from the linear transformation $X = W^\top Z + \varepsilon$ of a d-dimensional latent Gaussian distribution $Z \sim \mathcal{N}(\boldsymbol{0}, I)$, possibly with added noise $\varepsilon \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\, I)$ so that $\mathbb{P} \equiv \mathcal{N}(\boldsymbol{0}, W^\top W + \sigma^2\, I)$. Maximizing the likelihood of this Gaussian over the encoding and decoding parameter W again yields PCA as the optimal solution [362]. Hence, PCA assumes that the data live on a d-dimensional ellipsoid embedded in data space $\mathcal{X} \subseteq \mathbb{R}^D$. Standard PCA, therefore, provides an illustrative example for the connections between density estimation and reconstruction.

Linear PCA, of course, is limited to data encodings that can only exploit linear feature correlations. kPCA [3] introduced a nonlinear generalization of component analysis by extending the PCA objective to nonlinear kernel feature maps and taking advantage of the "kernel trick." For a PSD kernel $k(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ with feature map $\phi_k : \mathcal{X} \to \mathcal{F}_k$, kPCA solves the reconstruction objective (26) in feature space $\mathcal{F}_k$:

$$\min_{W} \sum_{i=1}^{n} \|\phi_k(\boldsymbol{x}_i) - W^\top W\phi_k(\boldsymbol{x}_i)\|^2 \quad \text{s.t. } WW^\top = I \tag{27}$$

which results in an eigenvalue problem of the kernel matrix [3]. For kPCA, the reconstruction error can again serve as an anomaly score. It can be computed implicitly via the dual [4]. This reconstruction from linear principal components in feature space $\mathcal{F}_k$ corresponds to a reconstruction from some nonlinear subspace or manifold in input space $\mathcal{X}$ [371]. Replacing the reconstruction $W^\top W\phi_k(\boldsymbol{x})$ in (27) with a prototype $\boldsymbol{c} \in \mathcal{F}_k$ yields a reconstruction model that considers the squared error to the kernel mean since the prototype is optimally solved by $\boldsymbol{c} = (1/n)\sum_{i=1}^{n}\phi(\boldsymbol{x}_i)$ for the L^2-distance. For RBF kernels, this prototype model is (up to a multiplicative constant) equivalent to KDE [4], which provides a link between kernel reconstruction and nonparametric density estimation methods. Finally, rPCA variants have been introduced as well [372]–[375], which account for data contamination or noise (see Section II-C2).

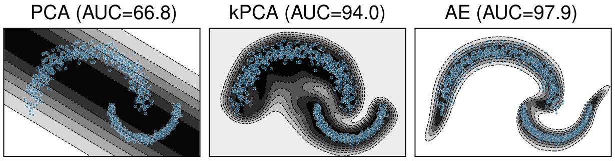


Fig. 8. *Reconstruction models on the Big Moon, Small Moon toy example (see Fig. 4). PCA finds the linear subspace with the lowest reconstruction error under an orthogonal projection of the data. kPCA solves (linear) component analysis in kernel feature space, which enables an optimal reconstruction from (kernel-induced) nonlinear components in input space. An AE with 1-D latent code learns a 1-D, nonlinear manifold in input space having minimal reconstruction error.*

C. Autoencoders

AEs are reconstruction models that use neural networks for the encoding and decoding of data. They were originally introduced during the 1980s [376]–[379] primarily as methods to perform nonlinear dimensionality reduction [380], [381], yet they have also been studied early on for AD [351], [352]. Today, deep AEs are among the most widely adopted methods for deep AD in the literature [44], [51], [54], [125]–[135] likely due to their long history and easy-to-use standard variants. The standard AE objective is given by

$$\min_{\omega} \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - (\phi_d \circ \phi_e)_\omega(\boldsymbol{x}_i)\|^2 + \mathcal{R} \tag{28}$$

which is a realization of the general reconstruction objective (23) with $\theta = \omega$, that is, the optimization is carried out over the weights ω of the neural network encoder and decoder. A common way to regularize AEs is by mapping to a lower dimensional "bottleneck" representation $\phi_e(\boldsymbol{x}) = \boldsymbol{z} \in \mathcal{Z}$ through the encoder network, which enforces data compression and effectively limits the dimensionality of the manifold or subspace to be learned. If linear networks are used, such an AE, in fact, recovers the same optimal subspace as spanned by the PCA eigenvectors [382], [383]. In Fig. 8, we show a comparison of three canonical reconstruction models (PCA, kPCA, and AE) trained on the *Big Moon, Small Moon* toy data set, each using a different feature representation (raw input, kernel, and neural network), resulting in different manifolds.

Apart from a "bottleneck," a number of different ways to regularize AEs have been introduced in the literature. Following ideas of sparse coding [384]–[387], sparse AEs [388], [389] regularize the (possibly higher dimensional, overcomplete) latent code toward sparsity, for example, via L^1 Lasso penalization [390]. DAEs [391], [392] explicitly feed noise-corrupted inputs $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon$ into the network, which is then trained to reconstruct the original inputs $\boldsymbol{x}$. DAEs, thus, provide a way to specify a noise model for ε (see Section II-C2), which has been applied for noise-robust acoustic novelty detection [42], for instance. In situations in which the training data are already corrupted with noise or unknown anomalies, robust deep AEs [127], which splits the data into well-represented and corrupted parts similar to rPCA [374], have been proposed. Contractive AEs (CAEs) [393] propose to penalize the Frobenius norm of the Jacobian of the encoder activations with respect to the inputs to obtain a smoother and more robust latent representation. Such ways of regularization influence the geometry and shape of the subspace or manifold that is learned by the AE, for example, by imposing some degree of smoothness or introducing invariances toward certain types of input corruptions or transformations [131]. Hence, these regularization choices should again reflect the specific assumptions of a given AD task.

Besides the above deterministic variants, probabilistic AEs have also been proposed, which again establish a connection to density estimation. The most explored class of probabilistic AEs are VAEs [272]–[274], as introduced in Section III-C1, through the lens of neural generative models, which approximately maximizes the data likelihood (or evidence) by maximizing the ELBO. From a reconstruction perspective, VAEs adopt a stochastic autoencoding process, which is realized by encoding and decoding the parameters of distributions (e.g., Gaussians) through the encoder and decoder networks, from which the latent code and reconstruction then can be sampled. For a standard Gaussian VAE, for example, where $q(\boldsymbol{z}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{x}}))$, $p(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{0}, I)$, and $p(\boldsymbol{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}}, I)$ with encoder $\phi_{e,\omega'}(\boldsymbol{x}) = (\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\sigma}_{\boldsymbol{x}})$ and decoder $\phi_{d,\omega}(\boldsymbol{z}) = \boldsymbol{\mu}_{\boldsymbol{z}}$, the empirical ELBO objective (10) becomes

$$\min_{\omega,\omega'} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{M}\Bigg[\frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{\mu}_{\boldsymbol{z}_{ij}}\|^2 + D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{z}_{ij};\boldsymbol{\mu}_{\boldsymbol{x}_i}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{x}_i}))\|\mathcal{N}(\boldsymbol{z}_{ij};\boldsymbol{0}, I))\Bigg] \tag{29}$$

where $\boldsymbol{z}_{i1}, \ldots, \boldsymbol{z}_{iM}$ are M Monte Carlo samples drawn from the encoding distribution $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$. Hence, such a VAE is trained to minimize the mean reconstruction error over samples from an encoded latent Gaussian that is regularized to be close to a standard isotropic Gaussian. VAEs have been used in various forms for AD [276], [278], [394], for instance, on multimodal sequential data with LSTMs in robot-assisted feeding [395] and for new physics mining at the Large Hadron Collider [74]. Another class of probabilistic AEs that has been applied to AD are AAEs [44], [51], [396]. By adopting an adversarial loss to regularize and match the latent encoding distribution, AAEs can employ any arbitrary prior $p(\boldsymbol{z})$, as long as sampling is feasible.

Finally, other AE variants that have been applied to AD include RNN-based AEs [194], [231], [397], [398], convolutional AEs [54], AE ensembles [126], [398], and variants that constrain the gradients [399] or actively control the latent code topology [400] of an AE. AEs also have been utilized in two-step approaches that use AEs for dimensionality reduction and apply traditional methods on the learned embeddings [136], [401], [402].

D. Prototypical Clustering

Clustering methods that make the prototype assumption provide another approach to reconstruction-based AD. As mentioned above, the reconstruction error here is usually given by the distance of a point to its nearest prototype, which ideally has been learned to represent a distinct mode of the normal data distribution. Prototypical clustering methods [403] include the well-known VQ algorithms k-means, k-medians, and k-medoids that define a Voronoi partitioning [404], [405] over the metric space where they are applied—typically the input space $\mathcal{X}$. Kernel variants of k-means have also been studied [406] and considered for AD [316]. GMMs with a finite number of k mixtures (see Section III-A) have been used for (soft) prototypical clustering as well. Here, the distance to each cluster (or mixture component) is given by the Mahalanobis distance that is defined by the covariance matrix of the respective Gaussian mixture component [261].

More recently, deep learning approaches to clustering have also been introduced [407]–[410], some also based on k-means [411], and adopted for AD [129], [401], [412]. As in deep one-class classification (see Section IV-D), a persistent question in deep clustering is how to effectively regularize against a feature map collapse [413]. Note that, while, for deep clustering methods, the reconstruction error is measured in latent space $\mathcal{Z}$, for deep AEs, it is measured in the input space $\mathcal{X}$ after decoding. Thus, a latent feature collapse (i.e., a constant encoder $\phi_e \equiv \boldsymbol{c} \in \mathcal{Z}$) would result in a constant decoding (the data mean at optimum) for an AE, which, generally, is a suboptimal solution of (28). For this reason, AEs seem less susceptible to a feature collapse though they have also been observed to converge to bad local optima under SGD optimization, specifically if they employ bias terms [137].

VI. UNIFYING VIEW OF ANOMALY DETECTION

In this section, we present a unifying view of the AD problem. We identify specific AD modeling components that allow us to characterize the many methods discussed above in a systematic way. Importantly, this view reveals connections that enable the transfer of algorithmic ideas between existing AD methods. Thus, it uncovers promising

Table 2 AD Methods Identified With Our Unifying View (Last Column Contains Representative References)

Method	Loss $\ell(s,y)$	Model $f_\theta(\boldsymbol{x})$	Feature Map $\phi(\boldsymbol{x})$		Parameter θ	Regularization $\mathcal{R}(f,\phi,\theta)$	Bayes?	References
Parametric Density	$-\log(s)$	$p(\boldsymbol{x} \mid \theta)$	$\boldsymbol{x}$	(input)	θ	choice of density class $\{p_\theta \mid \theta \in \Theta\}$	✗	[414], [415]
Gaussian/Mahalanobis	$-\log(s)$	$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu},\Sigma)$	$\boldsymbol{x}$	(input)	$(\boldsymbol{\mu},\Sigma)$	–	✗	[414], [415]
GMM	$-\log(s)$	$\sum_k \pi_k \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k,\Sigma_k)$	$\boldsymbol{x}$	(input)	$(\pi,\boldsymbol{\mu},\Sigma)$	number of mixture components K	latent	[416]
KDE	$-\log(s)$	$\exp(-\lVert\phi_k(\boldsymbol{x})-\boldsymbol{\mu}\rVert^2)$	$\phi_k(\boldsymbol{x})$	(kernel)	$\boldsymbol{\mu}$	kernel hyperparameters (e.g., bandwidth h)	✗	[255], [256]
EBMs	$-\log(s)$	$\frac{1}{Z(\theta)}\exp(-E(\phi(\boldsymbol{x}),\boldsymbol{z};\theta))$	$\phi(\boldsymbol{x})$	(various)	θ	latent prior $p(\boldsymbol{z})$	latent	[264], [271]
Normalizing Flows	$-\log(s)$	$p_{\boldsymbol{z}}(\phi_\omega^{-1}(\boldsymbol{x}))\,\lvert\det J_{\phi_\omega^{-1}}(\boldsymbol{x})\rvert$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	base distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$; diffeomorphism architecture	✗	[283], [288]
GAN (D-based)	$-\log(s)$	$\sigma(\langle\boldsymbol{w},\psi_\omega(\boldsymbol{x})\rangle)$	$\psi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\omega)$	adversarial training	✗	[56], [339]
Min. Vol. Sphere	$\max(0,s)$	$\lVert\boldsymbol{x}-\boldsymbol{c}\rVert^2-R^2$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c},R)$	νR^2	✗	[224]
Min. Vol. Ellipsoid	$\max(0,s)$	$(\boldsymbol{x}-\boldsymbol{c})^\top\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{c})-R^2$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c},R,\Sigma)$	$\nu(\frac{1}{2}\lVert\Sigma\rVert_{\text{Fr}}^2+R^2)$	✗	[307]
SVDD	$\max(0,s)$	$\lVert\phi_k(\boldsymbol{x})-\boldsymbol{c}\rVert^2-R^2$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{c},R)$	νR^2	✗	[7]
Semi-Sup. SVDD	$\max(0,ys)$	$\lVert\phi_k(\boldsymbol{x})-\boldsymbol{c}\rVert^2-R^2$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{c},R)$	νR^2	✗	[7], [229]
Soft Deep SVDD	$\max(0,s)$	$\lVert\phi_\omega(\boldsymbol{x})-\boldsymbol{c}\rVert^2-R^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c},R,\omega)$	νR^2; weight decay; collapse reg. (various)	✗	[137]
OC Deep SVDD	s	$\lVert\phi_\omega(\boldsymbol{x})-\boldsymbol{c}\rVert^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c},\omega)$	weight decay; collapse reg. (various)	✗	[137]
Deep SAD	s^y	$\lVert\phi_\omega(\boldsymbol{x})-\boldsymbol{c}\rVert^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c},\omega)$	weight decay	✗	[144]
OC-SVM	$\max(0,s)$	$\rho-\langle\boldsymbol{w},\phi_k(\boldsymbol{x})\rangle$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{w},\rho)$	$\nu(\frac{1}{2}\lVert\boldsymbol{w}\rVert^2-\rho)$	✗	[6]
OC-NN	$\max(0,s)$	$\rho-\langle\boldsymbol{w},\phi_\omega(\boldsymbol{x})\rangle$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\rho,\omega)$	$\nu(\frac{1}{2}\lVert\boldsymbol{w}\rVert^2-\rho)$; weight decay	✗	[328]
Bayesian DD	$\max(0,s)$	$\lVert\phi_k(\boldsymbol{x})-\boldsymbol{c}\rVert^2-R^2$	$\phi_k(\boldsymbol{x})$	(kernel)	$(\boldsymbol{c},R)$	$\boldsymbol{c}=\sum_i\alpha_i\phi_k(\boldsymbol{x}_i)$ with prior $\alpha\sim\mathcal{N}(\boldsymbol{\mu},\Sigma)$	fully	[323]
GT	$-\log(s)$	$\prod_k\sigma_k(\langle\boldsymbol{w},\phi_\omega(T_k(\boldsymbol{x}))\rangle)$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\omega)$	transformations $\mathcal{T}=\{T_1,\ldots,T_K\}$ for self-labeling	✗	[152], [153]
GOAD (CE)	$-\log(s)$	$\prod_k\sigma_k(-\lVert\phi_\omega(T_k(\boldsymbol{x}))-\boldsymbol{c}_k\rVert^2)$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K,\omega)$	transformations $\mathcal{T}=\{T_1,\ldots,T_K\}$ for self-labeling	✗	[155]
BCE (supervised)	$-y\log(s)-\frac{1-y}{2}\log(1-s)$	$\sigma(\langle\boldsymbol{w},\phi_\omega(\boldsymbol{x})\rangle)$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\omega)$	weight decay	✗	[332]
BNN (supervised)	$-y\log(s)-\frac{1-y}{2}\log(1-s)$	$\sigma(\langle\boldsymbol{w},\phi_\omega(\boldsymbol{x})\rangle)$	$\phi_\omega(\boldsymbol{x})$	(neural)	$(\boldsymbol{w},\omega)$	prior $p(\boldsymbol{w},\omega)$	fully	[417], [418]
PCA	s	$\lVert\boldsymbol{x}-W^\top W\boldsymbol{x}\rVert_2^2$	$\boldsymbol{x}$	(input)	W	$WW^\top=I$	✗	[365]
Robust PCA	s	$\lVert\boldsymbol{x}-W^\top W\boldsymbol{x}\rVert_1$	$\boldsymbol{x}$	(input)	W	$WW^\top=I$	✗	[372]
Probabilistic PCA	$-\log(s)$	$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{0},W^\top W+\sigma^2 I)$	$\boldsymbol{x}$	(input)	(W,σ^2)	linear latent Gauss model $\boldsymbol{x}=W^\top\boldsymbol{z}+\varepsilon$	latent	[362]
Bayesian PCA	$-\log(s)$	$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{0},W^\top W+\sigma^2 I)\,p(W \mid \boldsymbol{\alpha})$	$\boldsymbol{x}$	(input)	(W,σ^2)	linear latent Gauss model with prior $p(W \mid \boldsymbol{\alpha})$	fully	[363]
Kernel PCA	s	$\lVert\phi_k(\boldsymbol{x})-W^\top W\phi_k(\boldsymbol{x})\rVert^2$	$\phi_k(\boldsymbol{x})$	(kernel)	W	$WW^\top=I$	✗	[3], [4]
Autoencoder	s	$\lVert\boldsymbol{x}-\phi_\omega(\boldsymbol{x})\rVert_2^2$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	advers. (AAE), contract. (CAE), denois. (DAE), etc.	✗	[127], [135]
VAE	$-\log(s)$	$p_{\phi_\omega}(\boldsymbol{x} \mid \boldsymbol{z})$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	latent prior $p(\boldsymbol{z})$	latent	[274], [278]
GAN (G-based)	$-\log(s)$	$p_{\phi_\omega}(\boldsymbol{x} \mid \boldsymbol{z})$	$\phi_\omega(\boldsymbol{x})$	(neural)	ω	adversarial training and latent prior $p(\boldsymbol{z})$	latent	[50], [147]
k-means	s	$\lVert\boldsymbol{x}-\operatorname{argmin}_{\boldsymbol{c}_k}\lVert\boldsymbol{x}-\boldsymbol{c}_k\rVert_2\rVert_2^2$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K)$	number of prototypes K	✗	[403], [416]
k-medians	s	$\lVert\boldsymbol{x}-\operatorname{argmin}_{\boldsymbol{c}_k}\lVert\boldsymbol{x}-\boldsymbol{c}_k\rVert_1\rVert_1$	$\boldsymbol{x}$	(input)	$(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K)$	number of prototypes K	✗	[403]
VQ	s	$\lVert\boldsymbol{x}-\phi_d(\operatorname{argmin}_{\boldsymbol{c}_k}\lVert\phi_e(\boldsymbol{x})-\boldsymbol{c}_k\rVert)\rVert$	$\phi(\boldsymbol{x})$	(various)	$(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K)$	number of prototypes K	✗	[360], [361]

directions for future research, such as transferring concepts and ideas from kernel-based AD to deep methods, and vice versa.

A. Modeling Dimensions of the AD Problem

We identify the following five components or modeling dimensions for AD:

D1 **Loss**	$\ell:\mathbb{R}\times\mathcal{Y}\to\mathbb{R},(s,y)\mapsto\ell(s,y)$
D2 **Model**	$f_\theta:\mathcal{X}\to\mathbb{R},\boldsymbol{x}\mapsto f_\theta(\boldsymbol{x})$
D3 **Feature Map**	$\boldsymbol{x}\mapsto\phi(\boldsymbol{x})$
D4 **Regularization**	$\mathcal{R}(f,\phi,\theta)$
D5 **Inference Mode**	Frequentist or Bayesian $\theta\sim p(\theta)$

Dimension D1 **Loss** is the (scalar) loss function that is applied to the output of some model $f_\theta(\boldsymbol{x})$. Semisupervised or supervised methods use loss functions that incorporate labels, but, for the many unsupervised AD methods, we have $\ell(s,y) = \ell(s)$. D2 **Model** defines the specific model f_θ that maps an input $\boldsymbol{x}\in\mathcal{X}$ to some scalar value that is evaluated by the loss. We have arranged our previous three sections along this modeling dimension where we covered certain groups of methods that formulate models based on common principles, namely probabilistic modeling, one-class classification, and reconstruction. Due to the close link between AD and density estimation (see Section II-B5), many of the methods formulate a likelihood model $f_\theta(\boldsymbol{x}) = p_\theta(\boldsymbol{x} \mid \mathcal{D}_n)$ with negative log-loss $\ell(s) = -\log(s)$, that is, they have a negative log-likelihood objective, where $\mathcal{D}_n = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ denotes the training data. Dimension D3 captures the **Feature Map** $\boldsymbol{x}\mapsto\phi(\boldsymbol{x})$ that is used in a model f_θ. This could be an (implicit) feature map $\phi_k(\boldsymbol{x})$ defined by some given kernel k in kernel methods, for example, or an (explicit) neural network feature map $\phi_\omega(\boldsymbol{x})$ that is learned and parameterized with network weights ω in deep learning methods. With dimension D4 **Regularization**, we capture various forms of regularization $\mathcal{R}(f,\phi,\theta)$ of the model f_θ, the feature map ϕ, and their parameters θ in a broader sense. Note that θ here may include both model parameters and feature map parameters, that is, $\theta = (\theta_f,\theta_\phi)$, in general. θ_f could be the distributional parameters of a parametric density model, for instance, and θ_ϕ the weights of a neural network. Our last modeling dimension D5 describes the **Inference Mode**, specifically whether a method performs Bayesian inference [416].

The identification of the above modeling dimensions enables us to formulate a general AD learning objective that encompasses a broad range of AD methods:

$$\min_\theta \quad \frac{1}{n}\sum_{i=1}^{n}\ell(f_\theta(\boldsymbol{x}_i),y_i)+\mathcal{R}(f,\phi,\theta). \qquad (*)$$

Denoting the minimum of $(*)$ by θ^*, the anomaly score of a test input $\tilde{\boldsymbol{x}}$ is computed via the model $f_{\theta^*}(\tilde{\boldsymbol{x}})$. In the Bayesian case, where the objective in $(*)$ is the negative log-likelihood of a posterior $p(\theta \mid \mathcal{D}_n)$ induced by a prior distribution $p(\theta)$, we can predict in a fully Bayesian fashion via the expected model $\mathbb{E}_{\theta\sim p(\theta \mid \mathcal{D}_n)}f_\theta(\boldsymbol{x})$. In Table 2, we describe many well-known AD methods using our unifying view.

B. Comparative Discussion

In the following, we compare the various approaches in light of our unifying view and discuss how this view enables the transfer of concepts between existing AD methods. Table 2 shows that the probabilistic methods are largely based on the negative log-likelihood objective. The resulting negative log-likelihood anomaly scores provide a (usually continuous) ranking that is generally more informative than a binary density level set detector (see Section II-B7). Reconstruction methods provide such a ranking as well, with the anomaly score given by the difference of a data instance and its reconstruction under the model. Besides ranking and detecting anomalies, such scores make it possible to also rank inliers, which can be used, for example, to judge cluster memberships or determine prototypes (see Section V-D). Reconstruction is particularly well suited when the data follow some manifold or prototypical structure (see Section V-A). In comparison, standard one-class classification methods, which aim to estimate a discriminative level set boundary (see Section IV), usually do not rank inliers. This is typically incorporated into the learning objective via a hinge loss, as can be seen in Table 2. One-class classification is generally more sample-efficient and more robust to a nonrepresentative sampling of the normal data (e.g., a sampling bias toward specific normal modes) [224] but is consequentially also less informative. However, an inlier ranking for one-class classification can still be obtained via the distance of a point to the decision boundary, but such an approximate ranking may not faithfully represent in-distribution modes and so on. In addition to the theoretical comparison and discussion of AD methods in regard to our unifying view, we will present an empirical evaluation that includes methods from all three groups (probabilistic, one-class classification, and reconstruction) and three types of feature maps (raw input, kernel, and neural network) in Section VII-C, where we find that the detection performance in different data scenarios is very heterogeneous among the methods (with an advantage for deep methods on the more complex, semantic detection tasks). This exemplifies the fact that there is no simple "silver bullet" solution to the AD problem.

In addition to providing a framework for comparing methods, our unifying view also allows us to identify concepts that may be transferred between shallow and deep AD methods in a systematic manner. We discuss a few explicit examples to illustrate this point here. Table 2 shows that both the (kernel) SVDD and Deep SVDD employ a hypersphere model. This connection can be used to transfer adaptations of the hypersphere model from one world to another (from shallow to deep, or vice versa). The adoption of semisupervised [144], [229], [419] or multisphere [145], [155], [316] model extensions give successful examples for such a transfer. Next, note in Table 2, that deep AEs usually consider the reconstruction error in the original data space $\mathcal{X}$ after a neural network encoding and decoding. In comparison, kPCA defines the error in kernel feature space $\mathcal{F}_k$. One might ask whether using the reconstruction error in some neural feature space may also be useful for AEs, for instance, to shift detection toward higher level feature spaces. Recent work that includes the reconstruction error over the hidden layers of an AE [135], indeed, suggests that this concept can improve detection performance. Another question one might ask when comparing the reconstruction models in Table 2 is whether including the prototype assumption (see Section V-A2) could also be useful in deep autoencoding and how this can be achieved practically. The VQ-VAE model, which introduces a discrete codebook between the neural encoder and decoder, presents a way to incorporate this concept that has shown to result in reconstructions with improved quality and coherence in some settings [408], [409]. Besides these existing proofs of concept for transferring ideas, which we have motivated here from our unifying view, we outline further potential combinations to explore in future research in Section IX-A.

C. Distance-Based Anomaly Detection

Our unifying view focuses on AD methods that formulate some loss-based learning objectives. Apart from these methods, there also exists a rich literature on purely "distance-based" AD methods and algorithms that have been studied extensively in the data mining community, in particular. Many of these algorithms follow a lazy learning paradigm, in which there is no *a priori* training phase of learning a model, but, instead, new test points are evaluated with respect to the training instances only as they occur. We here group these methods as "distance-based" without further granularity but remark that various taxonomies for these types of methods have been proposed [161], [179]. Examples of such methods include nearest-neighbor-based methods [8], [9], [420]–[422], such as LOF [10] and partitioning tree-based methods [423], such as iForest [424], [425]. These methods usually also aim to capture the high-density regions of the data in some manner, for instance, by scaling distances in relation to local neighborhoods [10], and, thus, are most consistent with the formal AD problem definition presented in Section II. The majority of these algorithms have been studied and applied in the original input space $\mathcal{X}$. Few of them have been considered in the context of deep learning, but some hybrid AD approaches exist, which apply distance-based algorithms on top of deep neural feature maps from pre-trained networks (e.g., [426]).

VII. EVALUATION AND EXPLANATION

The theoretical considerations and unifying view above provide useful insights about the characteristics and underlying modeling assumptions of the different AD methods. What matters the most to the practitioner, however, is to evaluate how well an AD method performs on real data. In this section, we first present different aspects of evaluation, in particular, the problem of building a data set that includes meaningful anomalies, and the problem of robustly evaluating an AD model on the collected data. In the second step, we will look at the limitations of classical evaluation techniques, specifically their inability to directly inspect, and verify the exact strategy employed by some model for detection, for instance, which input variables that a model uses for prediction. We then present "XAI" approaches for enabling such deeper inspection of a model.

A. Building Anomaly Detection Benchmarks

Unlike standard supervised data sets, there is an intrinsic difficulty in building AD benchmarks: anomalies are

Table 3 Existing AD Benchmarks

k-classes-out	(Fashion-)MNIST, CIFAR-10, STL-10, ImageNet
Synthetic	MNIST-C [428], ImageNet-C [429], ImageNet-P [429], ImageNet-O [434]
Real-world	*Industrial:* MVTec-AD [190], PCB [435] *Medical:* CAMELYON16 [60], [436], NIH Chest X-ray [60], [437], MOOD [438], HCP/BRATS [51], Neuropathology [59], [124] *Security:* Credit-card-fraud [439], URL [440], UNSW-NB15 [441] *Time series:* NAB [442], Yahoo [443] *Misc.:* Emmott [433], ELKI [444], ODDS [445], UCI [446], [447]

rare, and some of them may have never been observed before they manifest themselves in practice. Existing anomaly benchmarks typically rely on one of the following strategies.

1) *k-classes-out:* Start from a binary or multiclass data set and declare one or more classes to be normal and the rest to be anomalous. Due to the semantic homogeneity of the resulting "anomalies," such a benchmark may not be a good simulacrum of real anomalies. For example, simple low-level anomalies (e.g., additive noise) may not be tested for.
2) *Synthetic:* Start from an existing supervised or unsupervised data set and generate synthetic anomalies (e.g., [427]–[429]). Having full control over anomalies is desirable from a statistical viewpoint, to get robust error estimates. However, the characteristics of real anomalies may be unknown or difficult to generate.
3) *Real-world:* Consider a data set that contains anomalies and have them labeled by a human expert. This is the ideal case. In addition to the anomaly label, the human can augment a sample with an annotation of which exact features are responsible for the anomaly (e.g., a segmentation mask in the context of image data).

We provide examples of AD benchmarks and data sets falling into these three categories in Table 3.

Although all three approaches are capable of producing anomalous data, we note that real anomalies may exhibit much wider and finer variations compared to those in the data set. In adversarial cases, anomalies may be designed maliciously to avoid detection (e.g., in fraud and cybersecurity scenarios [204], [347], [430]–[433]).

B. Evaluating Anomaly Detectors

Most applications come with different costs for false alarms (type I error) and missed anomalies (type II error). Hence, it is common to consider the decision function

$$\text{decide} \begin{cases} \text{anomaly}, & \text{if } s(\boldsymbol{x}) \geq \tau \\ \text{inlier}, & \text{if } s(\boldsymbol{x}) < \tau \end{cases} \tag{30}$$

where s denotes the anomaly score, and adjust the decision threshold τ in a way that 1) minimizes the costs associated with the type I and type II errors on the collected validation data or 2) accommodates the hard constraints of the environment in which the AD system will be deployed.

To illustrate this, consider an example in financial fraud detection: anomaly alarms are typically sent to a fraud analyst who must decide whether to open an investigation into the potentially fraudulent activity. There are, typically, a fixed number of analysts. Suppose they can only handle k alarms per day, that is, the k examples with the highest predicted anomaly score. In this scenario, the measure to optimize is the precision@k since we want to maximize the number of anomalies contained in those k alarms.

In contrast, consider a credit card company that places an automatic hold on a credit card when an anomaly alarm is reported. False alarms result in angry customers and reduced revenue, so the goal is to maximize the number of true alarms subject to a constraint on the percentage of false alarms. The corresponding measure is to maximize recall@k, where k is the number of false alarms.

However, it is often the case that application-related costs and constraints are not fully specified or vary over time. With such restrictions, it is desirable to have a measure that evaluates the performance of AD models under a broad range of possible application scenarios, or analogously, a broad range of decision thresholds τ. The AUROC (or simply AUC) provides an evaluation measure that considers the full range of decision thresholds on a given test set [448], [449]. The ROC curve plots all the (false alarm rate, recall)-pairs that result from iterating over all thresholds that cover every possible test set decision split, and the area under this curve is the AUC measure. A convenient property of the AUC is that the random guessing baseline always achieves an AUC of 0.5, regardless of whether there is an imbalance between anomalies and normal instances in the test set. This makes AUC easy to interpret and comparable over different application scenarios, which is one of the reasons why AUC is the most commonly used performance measure in AD [444], [450]. One caveat of AUC is that it can produce overly optimistic scores in the case of highly imbalanced test sets [200], [451]. In such cases, the AUPRC is more informative and appropriate to use [200], [451]. The PR curve plots all the (precision, recall)-pairs that result from iterating over all possible test set decision thresholds. AUPRC, therefore, is preferable to AUROC when precision is more relevant than the false alarm rate. A common robust way to compute AUPRC is via AP [452]. One downside of AUPRC (or AP) is that the random guessing baseline is given by the fraction of anomalies in the test set and, thus, varies between applications. This makes AUPRC (or AP) generally harder to interpret and less comparable over different application scenarios. In scenarios where there is no clear preference for precision or the false alarm rate, we recommend to ideally report both threshold-independent measures for a comprehensive evaluation.

Table 4 AUC Detection Performance on MNIST-C

	Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DOCC	AE
brightness	**100.0**	99.0	**100.0**	**100.0**	100.0	**100.0**	**100.0**	13.7	**100.0**
canny edges	99.4	68.4	**100.0**	78.9	96.3	99.9	100.0	97.9	100.0
dotted line	99.9	62.9	99.3	68.5	70.0	92.6	91.5	86.4	**100.0**
fog	100.0	89.6	98.1	62.1	92.3	91.3	**100.0**	17.4	100.0
glass blur	79.5	34.7	70.7	8.0	49.1	27.1	**100.0**	31.1	99.6
impulse noise	**100.0**	69.0	**100.0**	98.0	99.7	**100.0**	**100.0**	97.5	**100.0**
motion blur	38.1	43.4	24.3	8.1	50.2	18.3	**100.0**	70.7	95.1
rotate	31.3	54.7	24.9	37.1	57.7	38.7	**93.2**	65.5	53.4
scale	7.5	20.7	14.5	5.0	36.5	19.6	68.1	**79.8**	40.4
shear	63.7	58.1	55.5	49.9	58.2	54.1	**94.9**	64.6	70.6
shot noise	94.9	43.2	97.1	41.6	63.4	81.5	96.7	51.5	**99.7**
spatter	**99.8**	52.6	85.0	44.5	57.3	64.5	99.0	68.2	97.4
stripe	**100.0**	99.9	**100.0**	**100.0**	100.0	**100.0**	**100.0**	100.0	**100.0**
translate	94.5	73.9	96.3	76.2	91.8	94.8	97.3	**98.8**	92.2
zigzag	99.9	72.5	100.0	84.0	87.7	99.4	98.3	94.3	**100.0**

C. Comparison on MNIST-C and MVTec-AD

In the following, we apply the AUC measure to compare a selection of AD methods from the three major approaches (probabilistic, one-class, and reconstruction) and three types of feature representation (raw input, kernel, and neural network). We perform the comparison on the synthetic MNIST-C and real-world MVTec-AD data sets. MNIST-C is MNIST extended with a set of 15 types of corruptions (e.g., blurring, added stripes, and impulse noise). MVTec-AD consists of 15 image sets from industrial production, where anomalies correspond to manufacturing defects. These images sometimes take the form of textures (e.g., wood and grid) or objects (e.g., toothbrush and screw). For MNIST-C, models are trained on the standard MNIST training set and then tested on each corruption separately. We measure the AUC separating the corrupted from the uncorrupted test set. For MVTec-AD, we train distinct models on each of the 15 image sets and measure the AUC on the corresponding test set. Results for each model are shown in Tables 4 and 5. We provide the training details of each model in Appendix B-B.

The first striking observation is the heterogeneity in the performance of the various methods on the different corruptions and defect classes. For example, AGAN performs generally well on MNIST-C but is systematically outperformed by the deep one-class classification (DOCC) model on MVTec-AD. Also, the more powerful nonlinear models are not better in every class, and simple "shallow" models occasionally outperform their deeper counterparts. For instance, the simple Gaussian model reaches top performance on MNIST-C:spatter, linear PCA ranks highest on MVTec-AD:toothbrush, and KDE ranks highest on MVTec-AD:wood. The fact that some of the simplest models, sometimes, perform well highlights the strong differences in the modeling structure of each AD model.

Table 5 AUC Detection Performance on MVTec-AD

		Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DOCC	AE
Textures	carpet	48.8	63.5	45.6	34.8	48.7	41.9	83.1	**90.6**	36.8
	grid	60.6	67.8	81.8	71.7	80.4	76.7	**91.7**	52.4	74.6
	leather	39.6	49.5	60.3	41.5	57.3	61.1	58.6	**78.3**	64.0
	tile	68.5	79.7	56.4	68.9	73.3	63.2	74.1	**96.5**	51.8
	wood	54.0	80.1	90.4	**94.7**	94.1	90.6	74.5	91.6	88.5
Objects	bottle	78.9	67.0	97.4	83.3	89.3	96.3	90.6	**99.6**	95.0
	cable	56.5	71.9	77.6	66.9	73.1	75.6	69.7	**90.9**	57.3
	capsule	71.6	65.1	75.7	56.2	61.3	71.5	60.7	**91.0**	52.5
	hazelnut	67.6	80.4	89.1	69.9	74.3	83.8	**96.4**	95.0	90.5
	metal nut	54.7	45.1	56.4	33.3	54.3	59.0	79.3	**85.2**	45.5
	pill	65.5	71.5	**82.5**	69.1	76.2	80.7	64.6	80.4	76.0
	screw	53.5	35.5	67.9	36.9	8.6	46.7	**99.6**	86.9	77.9
	toothbrush	93.9	76.1	**98.3**	93.3	96.1	**98.3**	70.8	96.4	49.4
	transistor	70.2	64.8	81.8	72.4	74.8	80.0	78.8	**90.8**	51.2
	zipper	50.1	65.2	82.8	61.4	68.6	81.0	69.7	**92.4**	35.0

Since the MNIST-C and MVTec-AD test sets are not highly imbalanced, we see the same trends when using AP as an evaluation measure as to be expected [451]. We provide the detection performance results in AP in Appendix B-A.

However, what is still unclear is whether the measured model performance faithfully reflects the performance on a broader set of anomalies (i.e., the generalization performance) or whether some methods only benefit from the specific (possibly nonrepresentative) types of anomalies that have been collected in the test set. In other words, assuming that all models achieve 100% test accuracy (e.g., MNIST-C:stripe), can we conclude that all models will perform well on a broad range of anomalies? This problem has been already highlighted in the context of supervised learning, and explanation methods can be applied to uncover such potential hidden weaknesses of models, also known as "Clever Hanses" [250].

D. Explaining Anomalies

In the following, we consider techniques that augment anomaly predictions with explanations. These techniques enable us to better understand the generalization properties and detection strategies used by different anomaly models and, in turn, to also address some of the limitations of classical validation procedures. Producing explanations of model predictions is already common in supervised learning, and this field is often referred to as XAI [251]. Popular XAI methods include LIME [453], (guided) Grad-CAM [454], integrated gradients [455], [456], and layerwise relevance propagation (LRP) [457]. Grad-CAM and LRP rely on the structure of the network to produce robust explanations.

XAI has recently also been brought to unsupervised learning and, in particular, AD [38], [334], [337], [458]–[460]. Unlike supervised learning, which is largely dominated by neural networks [81], [84], [461], state-of-the-art methods for unsupervised learning are much more heterogeneous, including neural networks but also kernel-, centroid-, or probability-based models. In such a heterogeneous setting, it is difficult to build explanation methods that allow for a consistent comparison of detection strategies of the multiple AD models. Two directions to achieve such consistent explanations are particularly promising:

1) model-agnostic explanation techniques (e.g., sampling-based) that apply transparently to any model, whether it is a neural network or something different (e.g., [458]);
2) a conversion of non-neural network models into functionally equivalent neural networks, or neuralization, so that existing approaches for explaining neural networks (e.g., LRP [457]) can be applied [334], [460].

In the following, we demonstrate a neuralization approach. It has been shown that numerous AD models, in particular, kernel-based models, such as KDE or one-class SVMs, can be rewritten as strictly equivalent

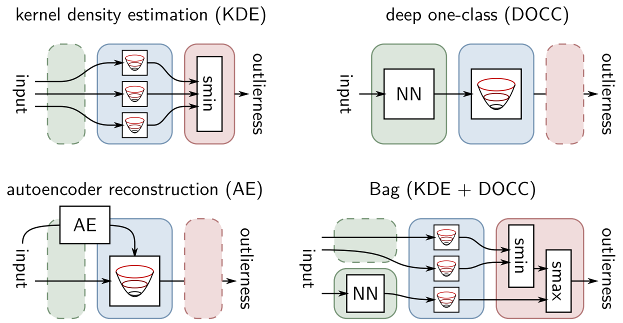


Fig. 9. ***Illustration of the neuralization concept that reformulates models as strictly equivalent neural networks. Here, KDE, deep one-class classification (DOCC), and AE are expressed as a three-layer architecture [334]: 1) feature extraction → 2) distance computation → and 3) pooling . The "neuralized" formulation enables to apply LRP [457] for explaining anomalies. A bag of models (here KDE and DOCC) can also be expressed in this way.***

neural networks [334], [460]. The neuralized equivalents of a model may not be unique, and explanations obtained with LRP consequently depend on the chosen network structure [462]. Here, we aim to find a single structure that fits many models. We show examples of neuralized models in Fig. 9. They typically organize into a three-layer architecture; from left to right: feature extraction, distance computation, and pooling.

For example, the KDE model, usually expressed as $f(\boldsymbol{x}) = (1/n)\sum_{i=1}^{n}\exp(-\gamma\,\|\boldsymbol{x}-\boldsymbol{x}_i\|^2)$, can have its negative log-likelihood $s(\boldsymbol{x}) = -\log f(\boldsymbol{x})$ rewritten as a two-layer network

$$h_j = \gamma\,\|\boldsymbol{x}-\boldsymbol{x}_j\|^2 + \log n \qquad \text{(layer 1)}$$
$$s(\boldsymbol{x}) = \text{smin}_j\{h_j\} \qquad \text{(layer 2)}$$

where smin is a soft min-pooling of the type log-sum-exp (see [460] for further details).

Once the model has been converted into a neural network, we can apply explanation techniques, such as LRP [457], to produce an explanation of the anomaly prediction. In this case, the LRP algorithm will take the score at the output of the model, propagate to "winners" in the pool, then assign the score to directions in the input or feature space that contribute the most to the distance, and, if necessary, propagate the signal further down the feature hierarchy (see the Supplement of [334] for how this is done exactly).

Fig. 10 shows, from left to right, an anomaly from the MNIST-C data set, the ground-truth explanation (the squared difference between the digit before and after corruption), and LRP explanations for three AD models (KDE, DOCC, and AE).

From these observations, it is clear that each model, although predicting with 100% accuracy on the current data, will have different generalization properties and vulnerabilities when encountering subsequent anomalies. We will work through an example in Section VIII-B to show how explanations can help to diagnose and improve a detection model.

To conclude, we emphasize that a standard quantitative evaluation can be imprecise or even misleading when the available data are not fully representative, and in that case, explanations can be produced to more comprehensively assess the quality of an AD model.

VIII. WORKED-THROUGH EXAMPLES

In this section, we work through two specific, real-world examples to exemplify the modeling and evaluation process and provide some best practices.

A. Example 1: Thyroid Disease Detection

In the first example, our goal is to learn a model to detect thyroid gland dysfunctions, such as hyperthyroidism. The Thyroid data set[4] includes $n = 3772$ data instances and has $D = 6$ real-valued features. It contains a total of 93 (∼2.5%) anomalies. For a quantitative evaluation, we consider a data set split of 60:10:30 corresponding to the training, validation, and test sets, respectively, while preserving the ratio of ∼2.5% anomalies in each of the sets.

We choose the OC-SVM [6] with standard RBF kernel $k(\boldsymbol{x},\tilde{\boldsymbol{x}}) = \exp(-\gamma\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|^2)$ as a method for this task since the data is real-valued and low-dimensional, and the OC-SVM scales sufficiently well for this comparatively small data set. In addition, the ν-parameter formulation [see (20)] enables us to use our prior knowledge and, thus, approximately control the false alarm rate α and, with it, implicitly also the miss rate, which leads to our first recommendation:

Assess the risks of false alarms and missed anomalies

Calibrating the false alarm rate and miss rate of a detection model can make the difference between life and death in a medical context, such as disease detection. Though the consequences must not always be as dramatic

[4] Available from the ODDS Library [445] at http://odds.cs.stonybrook.edu/

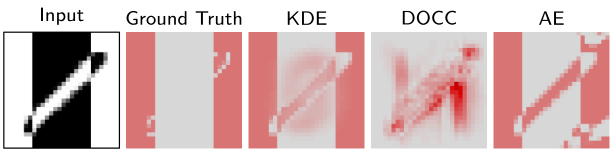


Fig. 10. ***Example of anomaly explanations. The input is an anomalous digit 1 from MNIST-C:stripe that has been corrupted by inverting the pixels in the left and right vertical stripes. The ground-truth explanation highlights the anomalous pixels in red. The KDE, DOCC, and AE detect the stripe anomalies accurately, but the LRP explanations show that the strategies are very different: KDE highlights the anomaly but also some regions of the digit itself. DOCC strongly emphasizes vertical edges. The AE produces a result similar to KDE but with decision artifacts in the corners of the image and on the digit itself.***

as in a medical setting, it is important to carefully consider the risks and costs involved with type I and type II errors in advance. In our example, a false alarm would suggest a thyroid dysfunction although the patient is healthy. On the other hand, a missed alarm would occur if the model recognizes a patient with dysfunction as healthy. Such asymmetric risks, with a greater expected loss for anomalies that go undetected, are very common in medical diagnosis [463]–[466]. Given only $D = 6$ measurements per data record, we, therefore, seek to learn a detector with a miss rate ideally close to zero, at the cost of an increased false alarm rate. Patients falsely ascribed with dysfunction by such a detector could then undergo further, more elaborate clinical testing to verify the disease. Assuming that our data are representative and ∼12%[5] of the population is at risk of thyroid dysfunction, we choose a slightly higher $\nu = 0.15$ to further increase the robustness against potential data contamination (here, the training set contains ∼2.5% contamination in the form of unlabeled anomalies). We then train the model and choose the kernel scale γ according to the best AUC we observe on the small, labeled validation set that includes nine labeled anomalies. We select γ from $\gamma \in \{(2^i D)^{-1} \,|\, i = -5, \ldots, 5\}$, that is, from a $\log_2$ span that accounts for the dimensionality D.

Following the above, we observe a rather poor best validation set AUC of 83.9% at $\gamma = (2^{-5}D)^{-1}$, which is the largest value from the hyperparameter range. This is an indication that we forgot an important preprocessing step, namely

Apply feature scaling to normalize value ranges

Any method that relies on computing distances, including kernel methods, requires the features to be scaled to similar ranges to prevent features with wider value ranges from dominating the distances. If this is not done, it can cause anomalies that deviate on smaller scale features to be undetected. Similar reasoning also holds for clustering and classification (e.g., see discussions in [467] or [468]). Min–max normalization or standardization is a common choice, but, since we assume there might be some contamination, we apply a robust feature scaling via the median and interquartile range. Remember that scaling parameters should be computed using only information from the training data and then applied to all of the data. After we have scaled the features, we observe a much improved best validation set AUC of 98.6% at $\gamma = (2^2 D)^{-1}$. The so-trained and selected model finally achieves a test set AUC of 99.2%, a false alarm rate of 14.8% (i.e., close to our *a priori* specified $\nu = 0.15$), and a miss rate of zero.

B. Example 2: MVTec Industrial Inspection

In our second example, we consider the task of detecting anomalies in wood images from the MVTec-AD data set. Unlike the first worked-through example, the MVTec data are high-dimensional and correspond to arrays of pixel values. Hence, all input features are already on a similar scale (between -1 and $+1$), and thus, we do not need to apply feature rescaling.

Table 6 AUC Detection Performance on the MVTec-AD "Wood" Class

Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DOCC	AE
54.0	80.1	90.4	**94.7**	94.1	90.6	74.5	91.6	88.5

Following the standard model training/validation procedure, we train a set of models on the training data, select their hyperparameters on hold out data (e.g., a few inliers and anomalies extracted from the test set), and then evaluate their performance on the remainder of the test set. Table 6 shows the AUC performance of the nine models in our benchmark.

We observe that the best-performing model is KDE. This is particularly surprising because this model does not compute the kinds of higher level image features that deep models, such as DOCC, learn, and apply. An examination of the data set shows that the anomalies involve properties, such as small perforations and stains that do not require high-level semantic information to be detected. But is that the only reason why the performance of KDE is so high? In order to get insights into the strategy used by KDE to arrive at its prediction, we employ the neuralization/LRP approach presented in Section VII-D.

Apply XAI to analyze model predictions

Fig. 11 shows an example of an image along with its ground-truth pixel-level anomaly and the computed pixelwise explanation for KDE.

Ideally, we would like the model to make its decision based on the actual anomaly (here, the liquid stain), and therefore, we would expect the ground-truth annotation and the KDE explanation to coincide. However, it is clear from an inspection of the explanation that KDE is not looking at the true cause of the anomaly and is looking instead at the vertical stripes present everywhere in the input image. This discrepancy between the explanation and the ground truth can be observed on other images of the "wood" class. The high AUC score of KDE, thus, must

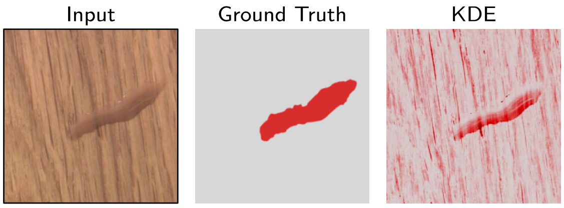


Fig. 11. ***Input image, ground-truth source of the anomaly (here, a stain of liquid), and the explanation of the KDE anomaly prediction. The KDE model assigns high relevance to the wood grain instead of the liquid stain. This discrepancy between the ground truth and model explanation reveals a "Clever Hans" strategy used by the KDE model.***

[5] https://www.thyroid.org/

be due to a spurious correlation in the test set between the reaction of the model to these stripes and the presence of anomalies. We call this a “Clever Hans” effect [250], because, just like the horse Hans, who could correctly answer arithmetic problems by reading unintended reactions of his master,[6] the model appears to work because of a spurious correlation. It is obvious that the KDE model is unlikely to generalize well when the anomalies and the stripes become decoupled (e.g., as we observe more data or under some adversarial manipulation). This illustrates the importance of generating explanations to identify these kinds of failures. Once we have identified the problem, how can we change our AD strategy so that it is more robust and generalizes better?

Improve the model based on explanations

In practice, there are various approaches to improve the model based on explanation feedback.

1) *Data extension:* We can extend the data with missing training cases, for instance, anomalous wood examples that lack stripes or normal wood examples that have stripes to break to a spurious correlation between stripes and anomalies. When further data collection is not possible, synthetic data extension schemes, such as blurring or sharpening, can also be considered.
2) *Model extension:* If the first approach is not sufficient, or if the model is simply not capable of implementing the necessary prediction structure, the model itself can be changed (e.g., using a more flexible deep model). In other cases, the model may have enough representation power but is statistically inefficient (e.g., subject to the curse of dimensionality). In that case, adding structure (e.g., convolutions) or regularization can also help to learn a model with an appropriate prediction strategy.
3) *Ensembles:* If all considered models have their own strengths and weaknesses, ensemble approaches can be considered. Ensembles have a conceptual justification in the context of AD [334], and they have been shown to work well empirically [469], [470].

Once the model has been improved using these strategies, explanations can be recomputed and examined to verify that the decision strategy has been corrected. If that is not the case, the process can be iterated until we reach a satisfactory model.

In our wood example, we have observed that KDE reacts strongly to the vertical strains. Based on this observation, we replace the Gaussian kernel with a Mahalanobis kernel that effectively applies a horizontal Gaussian blur to the images before computing the distance. This has the effect of reducing the strain patterns, but keeping the anomalies intact. This increases the explanation accuracy of the model from an average cosine similarity of 0.34 to 0.38 on the ground-truth explanations. Fig. 12 shows the explanation of the improved model. Implementation details can be found in Appendix B-C. The AUC drops to 87%, which corresponds to a more realistic estimate of the generalization abilities of the KDE model, which previously was biased by the spurious correlation.

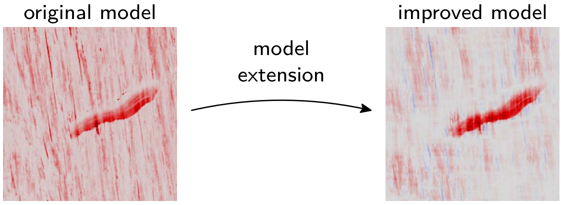


Fig. 12. ***Explanations of the original (left) and improved (right) KDE model. The Gaussian kernel strongly reacts to the vertical wood stripes (left). After replacing it with a Mahalanobis kernel (right) that is less sensitive to high horizontal frequencies, the model focuses on the true source of anomaly considerably better.***

IX. CONCLUSION AND OUTLOOK

AD is a blossoming field of broad theoretical and practical interest across the disciplines. In this work, we have given a review of the past and present state of AD research, established a systematic unifying view, and discussed many practical aspects. While we have included some of our own contributions, we hope that we have fulfilled our aim of providing a balanced and comprehensive snapshot of this exciting research field. The focus was given to a solid theoretical basis, which then allowed us to put today’s two main lines of development into perspective: the more classical kernel world and the more recent world of deep learning and representation learning for AD.

We will conclude our review by turning to what lies ahead. In the following, we highlight some critical open challenges—of which there are many—and identify a number of potential avenues for future research that we hope will provide useful guidance.

A. Unexplored Combinations of Modeling Dimensions

As can be seen in Fig. 1 and Table 2, there is a zoo of different AD algorithms that have historically been explored along various dimensions. This review has shown conceptual similarities between AD members from kernel methods and deep learning. Note, however, that the exploration of novel algorithms has been substantially different in both domains, which offers unique possibilities to explore new methodology: steps that have been pursued in kernel learning but not in deep AD could be transferred (or vice versa) and powerful new directions could emerge. In other words, ideas could be readily transferred from kernels to deep learning and back, and novel combinations in our unifying view would emerge.

Let us now discuss some specific opportunities to clarify this point. Consider the problem of robustness to noise and contamination or signal-to-noise ratio. For shallow methods, the problem is well studied, and we have many effective methods [5], [259], [326], [372], [374], [375], [471].

[6] https://en.wikipedia.org/wiki/Clever_Hans

In deep AD, very little work has addressed this problem. The second example is the application of Bayesian methods. The Bayesian inference has been mostly considered for shallow methods [323], [363], due to the prohibitive cost or intractability of exact Bayesian inference in deep neural networks. Recent progress in approximate Bayesian inference and Bayesian neural networks [418], [472]–[475] raise the possibility of developing methods that complement anomaly scores with uncertainty estimates or uncertainty estimates of their respective explanations [476]. In the area of SSAD, ideas have already been successfully transferred from kernel learning [224], [229] to deep methods [144] for one-class classification. However, probabilistic and reconstruction methods that can make use of labeled anomalies are less explored. For time-series AD [169], [195], [201]–[204], where forecasting (i.e., conditional density estimation) models are practical and widely deployed, semisupervised extensions of such methods could lead to significant improvements in applications in which some labeled examples are available (e.g., learning from failure cases in monitoring tasks). Concepts from density ratio estimation [477], noise contrast estimation [478], or coding theory [479] could lead to novel semisupervised methods in principled ways. Finally, active learning strategies for AD [346]–[349], which identify informative instances for labeling, have primarily only been explored for shallow detectors and could be extended to deep learning approaches.

This is a partial list of opportunities that we have noticed. Further analysis of our framework will likely expose additional directions for innovation.

B. Bridging Related Lines of Research on Robustness

Other recent lines of research on robust deep learning are closely related to AD or may even be interpreted as special instances of the problem. These include OOD detection, model calibration, uncertainty estimation, and adversarial examples or attacks. Bridging these lines of research by working out the nuances of the specific problem formulations can be insightful for connecting concepts and transferring ideas to jointly advance research.

A basic approach to creating robust classifiers is to endow them with the ability to reject input objects that are likely to be misclassified. This is known as the problem of classification with a reject option, and it has been studied extensively [480]–[486]. However, this work focuses on objects that fall near the decision boundary where the classifier is uncertain.

One approach to making the rejection decision is to calibrate the classification probabilities and then reject objects for which no class is predicted to have high probability following Chow's optimal rejection rule [481]. Consequently, many researchers have developed techniques for calibrating the probabilities of classifiers [473], [487]–[492] or Bayesian uncertainty quantification [417], [418], [472], [474], [475], [493].

Recent work has begun to address other reasons for rejecting an input object. OOD detection considers cases where the object is drawn from a distribution different from the training distribution $\mathbb{P}^+$ [241], [490], [492], [494]–[496]. From a formal standpoint, it is impossible to determine whether an input x is drawn from one of two distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ if both distributions have support at x. Consequently, the OOD problem reduces to determining whether x lies outside regions of high density in $\mathbb{P}^+$, which is exactly the AD problem that we have described in this review.

The second reason to reject an input object is that it belongs to a class that was not part of the training data. This is the problem of open set recognition. Such objects can also be regarded as being generated by a distribution $\mathbb{P}^-$, so this problem also fits within our framework and can be addressed with the algorithms described here. Nonetheless, researchers have developed a separate set of methods for open set recognition [240], [497]–[501], and an important goal for future research is to evaluate these methods from the AD perspective and to evaluate AD algorithms from the open set perspective.

In rejection, OOD, and open set recognition problems, there is an additional source of information that is not available in standard AD problems: the class labels of the objects. Hence, the learning task combines classification with AD. Formally, the goal is to train a classifier on labeled data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ with class labels $y \in \{1, \ldots, k\}$ while also developing some measure to decide whether an unlabeled test point $\tilde{\boldsymbol{x}}$ should be rejected (for any of the reasons listed above). The class label information tells us about the structure of $\mathbb{P}^+$ and allows us to model it as a joint distribution $\mathbb{P}^+ \equiv \mathbb{P}_{X,Y}$. Methods for rejection, OOD, and open set recognition all take advantage of this additional structure. Note that the labels y are different from the labels that mark normal or anomalous points in supervised or SSAD (see Section II-C).

Research on the unresolved and fundamental issue of adversarial examples and attacks [502]–[511] is related to AD as well. We may interpret adversarial attacks as extremely hard-to-detect OOD samples [473], as they are specifically crafted to target the decision boundary and confidence of a learned classifier. Standard adversarial attacks find a small perturbation δ for an input $\boldsymbol{x}$ so that $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \delta$ yields some class prediction desired by the attacker. For instance, a perturbed image of a dog may be indistinguishable from the original to the human's eye, yet the predicted label changes from "dog" to "cat." Note that such an adversarial example $\tilde{\boldsymbol{x}}$ still likely is (and probably should) be normal under the data marginal $\mathbb{P}_X$ (an imperceptibly perturbed image of a dog shows a dog after all!) but the pair $(\tilde{\boldsymbol{x}}, \text{"cat"})$ should be anomalous under the joint $\mathbb{P}_{X,Y}$ [242]. Methods for OOD detection have been found to also increase adversarial robustness [153], [473], [496], [512], [513], some of which model the class conditional distributions for detection [242], [495], for the reason just described.

The above highlights the connection of these lines of research toward the general goal of robust deep models. Thus, we believe that connecting ideas and concepts in these lines (e.g., the use of spherical models in both AD [137], [155] and OOD [512], [514]) may help them to advance together. Finally, the assessment of the robustness of neural networks and their fail-safe design and integration are topics of high practical relevance that have recently found their way in international standardization initiatives (see Section II-A). Beyond doubt, understanding the brittleness of deep networks (also in the context of their explanations [515]) will be critical for their adoption in AD applications that involve malicious attackers, such as fraudsters or network intruders.

C. Interpretability and Trustworthiness

Much of AD research has been devoted to developing new methods that improve detection accuracy. In most applications, however, accuracy alone is not sufficient [334], [516], and further criteria, such as interpretability [249], [517] and trustworthiness [475], [518], [519], are equally critical, as demonstrated in Sections VII and VIII. For researchers and practitioners alike [520], it is vital to understand the underlying reasons for how a specific AD model reaches a particular prediction. Interpretable, explanatory feedback enhances model transparency, which is indispensable for accountable decision-making [521], uncovering model failures, such as Clever Hans behavior [250], [334], and understanding model vulnerabilities that can be insightful for improving a model or system. This is especially relevant in safety-critical environments [522], [523]. Existing work on interpretable AD has considered finding subspaces of anomaly discriminative features [458], [524]–[528], deducing sequential feature explanations [459], using featurewise reconstruction errors [56], [190], employing fully convolutional architectures [337], and explaining anomalies via integrated gradients [38] or LRP [334], [460]. In relation to the vast body of literature though, research on interpretability and trustworthiness in AD has seen comparatively little attention. The fact that anomalies may not share similar patterns (i.e., their heterogeneity) poses a challenge for their explanation, which also distinguishes this setting from interpreting supervised classification models. Furthermore, anomalies might arise due to the presence of abnormal patterns but conversely also due to a lack of normal patterns. While, for the former case, an explanation that highlights the abnormal features is satisfactory, how should an explanation for missing features be conceptualized? For example, given the MNIST data set of digits, what should an explanation of an anomalous all-black image be? The matters of interpretability and trustworthiness get more pressing as the task and data become more complex. Effective solutions for complex tasks will necessarily require more powerful methods, for which explanations become generally harder to interpret. We, thus, believe that future research in this direction will be imperative.

D. Need for Challenging and Open Data Sets

Challenging problems with clearly defined evaluation criteria on publicly available benchmark data sets are invaluable for measuring progress and moving a field forward. The significance of the ImageNet database [529], together with corresponding competitions and challenges [530], for progressing computer vision and supervised deep learning in the last decade give a prime example of this. Currently, the standard evaluation practices in deep AD [130], [135], [137], [141], [144], [148], [152]–[155], [233], [531], OOD detection [241], [277], [490], [494]–[496], [532], [533], and open set recognition [240], [497]–[500] still extensively repurpose classification data sets by deeming some data set classes to be anomalous or considering in-distribution versus OOD data set combinations (e.g., training a model on Fashion-MNIST clothing items and regarding MNIST digits to be anomalous). Although these synthetic protocols have some value, it has been questioned how well they reflect real progress on challenging AD tasks [200], [332]. Moreover, we think the tendency that only a few methods seem to dominate most of the benchmark data sets in the work cited above is alarming since it suggests a bias toward evaluating only the upsides of newly proposed methods, yet often critically leaving out an analysis of their downsides and limitations. This situation suggests a lack of diversity in the current evaluation practices and the benchmarks being used. In the spirit of *all models are wrong* [534], we stress that more research effort should go into studying when and how certain models are wrong and behave like Clever Hanses. We need to understand the tradeoffs that different methods make. For example, some methods are likely making a tradeoff between detecting low-level versus high-level semantic anomalies (see Section II-B2 and [200]). The availability of more diverse and challenging data sets would be of great benefit in this regard. Recent data sets, such as MVTec-AD [190], and competitions, such as the Medical Out-of-Distribution Analysis Challenge [438], provide excellent examples, but the field needs many more challenging open data sets to foster progress.

E. Weak Supervision and Self-Supervised Learning

The bulk of AD research has been studying the problem in the absence of any kind of supervision, that is, in an unsupervised setting (see Section II-C2). Recent work suggests, however, that significant performance improvements on complex detection tasks seem achievable through various forms of weak supervision and self-supervised learning.

Weak supervision or weakly supervised learning describes learning from imperfectly or scarcely labeled data [535]–[537]. Labels might be inaccurate (e.g., due to labeling errors or uncertainty) or incomplete (e.g., covering only a few normal modes or specific anomalies). Current work on SSAD indicates that including even only few labeled anomalies can already yield remarkable performance improvements on complex data [60], [144], [332], [336], [337], [538]. A key challenge here is to formulate and optimize such methods so that they generalize well to novel anomalies. Combining these semisupervised methods with active learning techniques helps us to identify informative candidates for labeling [346]–[349]. It is an effective strategy for designing AD systems that continuously improve via expert feedback loops [459], [539]. This approach has not yet been explored for deep detectors, though. OE [233], that is, using massive amounts of data that is publicly available in some domains (e.g., stock photos for computer vision or the English Wikipedia for NLP) as auxiliary negative samples (see Section IV-E), can also be viewed as a form of weak supervision (imperfectly labeled anomalies). Although such negative samples may not coincide with ground-truth anomalies, we believe such contrast can be beneficial for learning characteristic representations of normal concepts in many domains (e.g., using auxiliary log data to well characterize the normal logs of a specific computer system [540]). So far, this has been little explored in applications. Transfer learning approaches to AD also follow the idea of distilling more domain knowledge into a model, for example, through using and possibly fine-tuning pretrained (supervised) models [139], [142], [334], [426], [541]. Overall, weak forms of supervision or domain priors may be essential for achieving effective solutions in semantic AD tasks that involve high-dimensional data, as has also been found in other unsupervised learning tasks, such as disentanglement [210], [542], [543]. Hence, we think that developing effective methods for weakly supervised AD will contribute to advancing the state of the art.

Self-supervised learning describes the learning of representations through solving auxiliary tasks, for example, next sentence and masked words prediction [111], future frame prediction in videos [544], or the prediction of transformations applied to images [545], such as colorization [546], cropping [547], [548], or rotation [549]. These auxiliary prediction tasks do not require (ground-truth) labels for learning and can, thus, be applied to unlabeled data, which makes self-supervised learning particularly appealing for AD. Self-supervised methods that have been introduced for visual AD train multiclass classification models based on pseudolabels that correspond to various geometric transformations (e.g., flips, translations, and rotations) [152]–[154]. An anomaly score can then be derived from the softmax activation statistics of a so-trained classifier, assuming that a high prediction uncertainty (close to a uniform distribution) indicates anomalies. These methods have shown significant performance improvements on the common k-classes-out image benchmarks (see Table 3). Bergman and Hoshen [155] have recently proposed a generalization of this idea to nonimage data, called GOAD, which is based on random affine transformations. We can identify GOAD and self-supervised methods based on geometric transformations (GT) as classification-based approaches within our unifying view (see Table 2). Other recent and promising self-supervised approaches are based on contrastive learning [156], [545], [550]. In a broader context, the interesting question will be to what extent self-supervision can facilitate the learning of semantic representations. There is some evidence that self-supervised learning helps improve the detection of semantic anomalies and, thus, exhibits inductive biases toward semantic representations [200]. On the other hand, there also exists evidence showing that self-supervision mainly improves learning of effective feature representations for low-level statistics [551]. Hence, this research question remains to be answered but bears great potential for many domains where large amounts of unlabeled data are available.

F. Foundation and Theory

The recent progress in AD research has also raised more fundamental questions. These include open questions about the OOD generalization properties of various methods presented in this review, the definition of anomalies in high-dimensional spaces, and information-theoretic interpretations of the problem.

Nalisnick *et al.* [277] have recently observed that DGMs (see Section III), such as normalizing flows, VAEs, or autoregressive models, can often assign a higher likelihood to anomalies than to in-distribution samples. For example, models trained on Fashion-MNIST clothing items can systematically assign a higher likelihood to MNIST digits [277]. This counterintuitive finding, which has been replicated in subsequent work [149], [233], [267], [532], [533], [552], revealed that there is a critical lack of theoretical understanding of these models. Solidifying evidence [243], [290], [532], [533] indicates that one reason seems to be that the likelihood in current DGMs is still largely biased toward low-level background statistics. Consequently, simpler data points attain higher likelihood (e.g., MNIST digits under models trained on Fashion-MNIST, but not vice versa). Another critical remark in this context is that, for (truly) high-dimensional data, the region with the highest density must not necessarily coincide with the region of highest probability mass (called the typical set), that is, the region where data points most likely occur [552]. For instance, while the highest density of a D-dimensional standard Gaussian distribution is given at the origin, points sampled from the distribution concentrate around an annulus with radius $\sqrt{D}$ for large D [553]. Therefore, points close to the origin have high density but are unlikely to occur. This mismatch questions the standard theoretical density (level set) problem formulation (see Section II-B) and use of likelihood-based anomaly detectors for some settings. Hence, theoretical research aimed at understanding the above phenomenon and DGMs, themselves, presents an exciting research opportunity.

Similar observations suggest that reconstruction-based models can systematically well reconstruct simpler OOD points that sit within the convex hull of the data.

Table 7 Notation Conventions

Symbol	Description
$\mathbb{N}$	The natural numbers
$\mathbb{R}$	The real numbers
D	The input data dimensionality $D \in \mathbb{N}$
$\mathcal{X}$	The input data space $\mathcal{X} \subseteq \mathbb{R}^D$
$\mathcal{Y}$	The labels $\mathcal{Y} = \{\pm 1\}$ ($+1$: normal; -1 : anomaly)
$\boldsymbol{x}$	A vector, e.g. a data point $\boldsymbol{x} \in \mathcal{X}$
$\mathcal{D}_n$	An unlabeled dataset $\mathcal{D}_n = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ of size n
$\mathbb{P}, p$	The data-generating distribution and pdf
$\mathbb{P}^+, p^+$	The normal data distribution and pdf
$\mathbb{P}^-, p^-$	The anomaly distribution and pdf
$\hat{p}$	An estimated pdf
ε	An error or noise distribution
$\mathrm{supp}(p)$	The support of a data distribution $\mathbb{P}$ with density p, i.e. $\{\boldsymbol{x} \in \mathcal{X} \,\vert\, p(\boldsymbol{x}) > 0\}$
$\mathcal{A}$	The set of anomalies
C_α	An α-density level set
$\hat{C}_\alpha$	An α-density level set estimator
τ_α	The threshold $\tau_\alpha \geq 0$ corresponding to C_α
$c_\alpha(\boldsymbol{x})$	The threshold anomaly detector corresponding to C_α
$s(\boldsymbol{x})$	An anomaly score function $s : \mathcal{X} \to \mathbb{R}$
$\mathbb{1}_A(\boldsymbol{x})$	The indicator function for some set A
$\ell(s, y)$	A loss function $\ell : \mathbb{R} \times \{\pm 1\} \to \mathbb{R}$
$f_\theta(\boldsymbol{x})$	A model $f_\theta : \mathcal{X} \to \mathbb{R}$ with parameters θ
$k(\boldsymbol{x}, \tilde{\boldsymbol{x}})$	A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
$\mathcal{F}_k$	The RKHS or feature space of kernel k
$\phi_k(\boldsymbol{x})$	The feature map $\phi_k : \mathcal{X} \to \mathcal{F}_k$ of kernel k
$\phi_\omega(\boldsymbol{x})$	A neural network $\boldsymbol{x} \mapsto \phi_\omega(\boldsymbol{x})$ with weights ω

Table 8 AP Detection Performance on MNIST-C

	Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DOCC	AE
brightness	**100.0**	98.0	**100.0**	**100.0**	100.0	**100.0**	**100.0**	32.9	**100.0**
canny edges	99.1	58.8	**100.0**	71.8	96.6	99.9	100.0	97.7	100.0
dotted line	99.9	56.8	99.0	63.4	67.9	90.9	88.8	81.5	**99.9**
fog	100.0	88.3	98.7	75.5	94.2	94.2	**100.0**	34.8	100.0
glass blur	78.6	42.0	65.5	31.5	45.9	36.2	**100.0**	37.6	99.6
impulse noise	**100.0**	59.8	**100.0**	97.1	99.6	**100.0**	**100.0**	96.2	**100.0**
motion blur	52.6	44.3	37.3	31.5	47.1	33.9	**100.0**	66.5	93.8
rotate	44.1	52.2	38.3	42.3	56.3	43.5	**93.6**	66.0	53.1
scale	31.9	34.5	33.0	31.2	39.4	34.4	61.9	**70.2**	42.5
shear	72.7	62.0	64.2	52.5	59.0	60.0	**95.5**	66.5	70.4
shot noise	93.6	44.8	97.3	42.7	60.4	81.7	96.8	49.0	**99.7**
spatter	**99.8**	50.5	82.6	45.8	54.8	61.2	99.2	63.2	97.1
stripe	**100.0**	99.9	**100.0**	**100.0**	100.0	**100.0**	**100.0**	100.0	**100.0**
translate	95.5	64.8	97.0	73.7	92.2	95.7	97.2	**98.6**	93.7
zigzag	99.8	64.6	100.0	79.4	86.5	99.3	98.0	94.8	**100.0**

Table 9 AP Detection Performance on MVTec-AD

		Gaussian	MVE	PCA	KDE	SVDD	kPCA	AGAN	DOCC	AE
Textures	carpet	77.3	86.9	71.0	70.2	77.4	69.8	94.3	**97.2**	70.9
	grid	79.9	80.8	91.7	85.5	89.2	88.7	**97.4**	75.4	84.8
	leather	72.9	81.1	85.8	75.3	83.6	86.3	82.1	**92.3**	87.7
	tile	84.4	91.6	80.5	85.1	86.9	83.9	88.8	**98.6**	78.1
	wood	82.0	93.8	97.0	**98.5**	98.3	97.1	92.0	97.6	96.8
Objects	bottle	92.3	86.2	99.2	94.2	96.7	98.9	97.2	**99.9**	98.5
	cable	73.2	76.6	85.9	78.5	82.9	84.2	81.2	**94.1**	71.3
	capsule	92.3	89.3	93.0	85.9	88.7	92.0	84.3	**97.9**	82.8
	hazelnut	81.9	89.3	94.2	83.2	85.7	90.9	**98.1**	97.5	95.0
	metal nut	86.3	82.6	86.5	75.0	86.0	87.4	92.7	**96.3**	77.0
	pill	91.8	93.8	**96.5**	91.7	95.0	96.1	90.6	95.6	94.5
	screw	78.0	71.4	86.6	69.1	55.4	77.0	**99.8**	95.1	90.3
	toothbrush	97.6	87.6	**99.4**	97.4	98.5	**99.4**	86.9	98.7	73.9
	transistor	70.5	54.7	80.7	70.1	74.1	79.7	71.2	**90.0**	51.4
	zipper	81.0	84.2	91.8	82.8	87.9	91.5	85.7	**97.8**	79.3

For example, an anomalous all-black image can be well reconstructed by an AE trained on MNIST digits [554]. An even simpler example is the perfect reconstruction of points that lie within the linear subspace spanned by the principal components of a PCA model, even in regions far away from the normal training data (e.g., along with the principal component in Fig. 8). While such OOD generalization properties might be desirable for general representation learning [555], such behavior critically can be undesirable for AD. Therefore, we stress that more theoretical research on understanding such OOD generalization properties or biases, especially for more complex models, will be necessary.

Finally, the push toward deep learning also presents new opportunities to interpret and analyze the AD problem from different theoretical angles. AEs, for example, can be understood from an information-theoretic perspective [556] as adhering to the Infomax principle [557]–[559] by implicitly maximizing the mutual information between the input and latent code—subject to structural constraints or regularization of the code (e.g., "bottleneck," latent prior, and sparsity)—via the reconstruction objective [391]. Similarly, information-theoretic perspectives of VAEs have been formulated showing that these models can be viewed as making a rate-distortion tradeoff [560] when balancing the latent compression (negative rate) and reconstruction accuracy (distortion) [561], [562]. This view has recently been used to draw a connection between VAEs and Deep SVDD, where the latter can be seen as a special case that only seeks to minimize the rate (maximize compression) [563]. Overall, AD has been studied comparatively less from an information-theoretic perspective [564], [565], yet we think this could be fertile ground for building a better theoretical understanding of representation learning for AD.

In conclusion, we firmly believe that AD in all its exciting variants will also in the future remain an indispensable practical tool in the quest to obtain robust learning models that perform well on complex data.

APPENDIX A NOTATION AND ABBREVIATIONS

For reference, we provide the notation and abbreviations used in this work in Table 7 and Nomenclature, respectively.

APPENDIX B ADDITIONAL DETAILS ON EXPERIMENTAL EVALUATION

A. Average Precision on MNIST-C and MVTec-AD

We provide the detection performance measured in AP of the experimental evaluation on MNIST-C and MVTec-AD from Section VII-C in Tables 8 and 9, respectively. As can be seen (and as to be expected [451]), the performance in AP here shows the same trends as AUC (see Tables 4 and 5) since the MNIST-C and MVTec-AD test sets are not highly imbalanced.

B. Training Details

For PCA, we compute the reconstruction error while maintaining 90% of the variance of the training data. We do the same for kPCA and additionally choose the kernel width such that 50% neighbors capture 50% of total similarity scores. For MVE, we use the fast MCD estimator [307] with a default support fraction of 0.9 and a contamination rate parameter of 0.01. To facilitate MVE computation on MVTec-AD, we first reduce the dimensionality via PCA retaining 90% of variance. For KDE, we choose the bandwidth parameter to maximize the likelihood of a small hold-out set from the training data. For SVDD, we consider $\nu \in \{0.01, 0.05, 0.1, 0.2\}$ and select the kernel scale using a small labeled hold-out set. The deep one-class classifier applies a whitening transform on the representations after the first fully connected layer of a pretrained VGG16 model (on MVTec-AD) or a CNN classifier trained on the EMNIST letter subset (on MNIST-C). For the AE on MNIST-C, we use a LeNet-type encoder that has two convolutional layers with max-pooling followed by two fully connected layers that map to an encoding of 64 dimensions and construct the decoder symmetrically. On MVTec-AD, we use an encoder–decoder architecture, as presented in [131], which maps to a bottleneck of 512 dimensions. Both the encoder and decoder here consist of four blocks having two 3×3 convolutional layers followed by max-pooling or upsampling, respectively. We train the AE such that the reconstruction error of a small training hold-out set is minimized. For AGAN, we use the AE encoder and decoder architecture for the discriminator and generator networks, respectively, where we train the GAN until convergence to a stable equilibrium.

C. Explaining KDE

The model can be neutralized as described in Section VII-D, replacing the squared Euclidean distance in the first layer with a squared Mahalanobis distance. The heatmaps of both models (KDE and Mahalanobis KDE) are computed as

$$R = \frac{1}{2}\sum_{j=1}^{n}(\boldsymbol{x}_j - \boldsymbol{x}) \odot \nabla_{\boldsymbol{x}_j} s(\boldsymbol{x})$$

where $\odot$ denotes elementwise multiplication. This implements a Taylor-type decomposition, as described in [460].

D. Open-Source Software, Tutorials, and Demos

For the implementation of the shallow MVE and SVDD models, we have used the `scikit-learn` library [566] available at https://scikit-learn.org/. For the implementation of the shallow Gauss, PCA, KDE, and kPCA models as well as the deep AGAN, DOCC, RealNVP, and AE models, we have used the `PyTorch` library [567] available at https://pytorch.org/. Implementations of the Deep SVDD and Deep SAD methods are available at https://github.com/lukasruff/. Tutorials, demos, and code for XAI techniques, in particular, LRP, can be found at http://www.heatmapping.org/. In the spirit of the need for open-source software in machine learning [568], a similar collection of tutorials, demos, and code on AD methods is in the making and will be made available at http://www.pyano.org/. ■

Acknowledgment

The authors kindly thank the reviewers and the editor for their constructive feedback that helped to improve this work.

REFERENCES

[1] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

[2] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441, 1933.

[3] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[4] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognit.*, vol. 40, no. 3, pp. 863–874, Mar. 2007.

[5] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, 2nd ed. Hoboken, NJ, USA: Wiley, 2009.

[6] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[7] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[8] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *VLDB J.*, vol. 8, no. 3, pp. 237–253, 2000.

[9] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 427–438.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

[11] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Statist.*, vol. 27, no. 3, pp. 832–837, Sep. 1956.

[12] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962.

[13] F. Y. Edgeworth, "On discordant observations," *Phil. Mag. J. Sci.*, vol. 23, no. 5, pp. 364–375, 1887.

[14] T. S. Kuhn, *The Structure of Scientific Revolutions*. Chicago, IL, USA: Univ. of Chicago Press, 1970.

[15] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007.

[16] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[17] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016.

[18] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 10, pp. 1–13, Jan. 2017.

[19] Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[20] R. K. Malaiya, D. Kwon, J. Kim, S. C. Suh, H. Kim, and I. Kim, "An empirical evaluation of deep learning for network anomaly detection," in *Proc. Int. Conf. Comput., Netw. Commun.*, Mar. 2018, pp. 893–898.

[21] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Stat. Sci.*, vol. 17, no. 3, pp. 235–255, 2002.

[22] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, Feb. 2011.

[23] H. Joudaki *et al.*, "Using data mining to detect health care fraud and abuse: A review of literature," *Global J. Health Sci.*, vol. 7, no. 1, pp. 194–202, Aug. 2014.

[24] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Gener. Comput. Syst.*, vol. 55, pp. 278–288, Feb. 2016.

[25] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016.

[26] G. van Capelleveen, M. Poel, R. M. Mueller, D. Thornton, and J. van Hillegersberg, "Outlier detection in healthcare fraud: A case study in the medicaid dental domain," *Int. J. Accounting Inf. Syst.*, vol. 21, pp. 18–31, Jun. 2016.

[27] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, and S.-Y. Chen, "Generative adversarial network based telecom fraud detection at the receiving bank," *Neural Netw.*, vol. 102, pp. 78–86, Jun. 2018.

[28] J. Rabatel, S. Bringay, and P. Poncelet, "Anomaly detection in monitoring sensor data for preventive maintenance," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7003–7015, Jun. 2011.

[29] J. Marzat, H. Piet-Lahanier, F. Damongeot, and E. Walter, "Model-based fault diagnosis for aerospace systems: A survey," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 226, no. 10, pp. 1329–1360, Oct. 2012.

[30] L. Martí, N. Sanchez-Pi, J. Molina, and A. Garcia, "Anomaly detection based on sensor data in

petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, Jan. 2015.

[31] W. Yan and L. Yu, "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, vol. 6, 2015. [Online]. Available: https://www.phmsociety.org/node/1652

[32] F. Lopez *et al.*, "Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1885–1892, Oct. 2017.

[33] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 387–395.

[34] D. J. Atha and M. R. Jahanshahi, "Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection," *Structural Health Monitor.*, vol. 17, no. 5, pp. 1110–1128, Sep. 2018.

[35] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, Aug. 2018.

[36] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.

[37] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high performance computing systems," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9428–9433.

[38] J. Sipple, "Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4368–4377.

[39] K. Golmohammadi and O. R. Zaiane, "Time series contextual anomaly detection for detecting market manipulation in stock market," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal.*, Oct. 2015, pp. 1–10.

[40] K. Golmohammadi and O. R. Zaiane, "Sentiment analysis on twitter to improve time series contextual anomaly detection for detecting stock market manipulation," in *Proc. Int. Conf. Big Data Anal. Knowl. Discovery*, 2017, pp. 327–342.

[41] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using one-class SVMs and wavelets for audio surveillance," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 763–775, Dec. 2008.

[42] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2015, pp. 1996–2000.

[43] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1d convolutional recurrent neural networks," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2017, pp. 80–84.

[44] E. Principi, F. Vesperini, S. Squartini, and F. Piazza, "Acoustic novelty detection with adversarial autoencoders," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 3324–3330.

[45] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the Neyman–Pearson lemma," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 1, pp. 212–224, Jan. 2019.

[46] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady, "Novelty detection for the identification of masses in mammograms," in *Proc. 4th Int. Conf. Artif. Neural Netw.*, 1995, pp. 442–447.

[47] S. Chauhan and L. Vig, "Anomaly detection in ECG time signals via deep long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal.*, Oct. 2015, pp. 1–7.

[48] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Sci. Rep.*, vol. 7, no. 1, pp. 1–14, Dec. 2017.

[49] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[50] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.

[51] X. Chen and E. Konukoglu, "Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders," in *Proc. Med. Imag. Deep Learn.*, 2018. [Online]. Available: https://openreview.net/forum?id=H1nGLZ2oG

[52] D. K. Iakovidis, S. V. Georgakopoulos, M. Vasilakakis, A. Koulaouzidis, and V. P. Plagianakos, "Detecting and locating gastrointestinal anomalies using deep learning and iterative cluster unification," *IEEE Trans. Med. Imag.*, vol. 37, no. 10, pp. 2196–2210, Oct. 2018.

[53] S. Latif, M. Usman, R. Rana, and J. Qadir, "Phonocardiographic sensing using deep learning for abnormal heartbeat detection," *IEEE Sensors J.*, vol. 18, no. 22, pp. 9393–9400, Nov. 2018.

[54] N. Pawlowski *et al.*, "Unsupervised lesion detection in brain CT using Bayesian convolutional autoencoders," in *Proc. Med. Imag. Deep Learn.*, 2018. [Online]. Available: https://openreview.net/forum?id=S1hpzoisz

[55] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Fusing unsupervised and supervised deep learning for white matter lesion segmentation," in *Proc. Med. Imag. Deep Learn.*, 2019, pp. 63–72.

[56] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "F-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Med. Image Anal.*, vol. 54, pp. 30–44, May 2019.

[57] P. Seeböck *et al.*, "Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal OCT," *IEEE Trans. Med. Imag.*, vol. 39, no. 1, pp. 87–98, Jan. 2019.

[58] P. Guo *et al.*, "Ensemble deep learning for cervix image selection toward improving reliability in automated cervical precancer screening," *Diagnostics*, vol. 10, no. 7, p. 451, 2020.

[59] L. Naud and A. Lavin, "Manifolds for unsupervised visual anomaly detection," 2020, *arXiv:2006.11364*. [Online]. Available: http://arxiv.org/abs/2006.11364

[60] N. Tuluptceva, B. Bakker, I. Fedulova, H. Schulz, and D. V. Dylov, "Anomaly detection with deep perceptual autoencoders," 2020, *arXiv:2006.13265*. [Online]. Available: http://arxiv.org/abs/2006.13265

[61] W.-K. Wong, A. W. Moore, G. F. Cooper, and M. M. Wagner, "Bayesian network anomaly pattern detection for disease outbreaks," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 808–815.

[62] W.-K. Wong, A. Moore, G. Cooper, and M. Wagner, "What's strange about recent events (WSARE): An algorithm for the early detection of disease outbreaks," *J. Mach. Learn. Res.*, vol. 6, pp. 1961–1998, Dec. 2005.

[63] R. Blender, K. Fraedrich, and F. Lunkeit, "Identification of cyclone-track regimes in the North Atlantic," *Quart. J. Roy. Meteorological Soc.*, vol. 123, no. 539, pp. 727–741, Apr. 1997.

[64] J. Verbesselt, A. Zeileis, and M. Herold, "Near real-time disturbance detection using satellite image time series," *Remote Sens. Environ.*, vol. 123, pp. 98–108, Aug. 2012.

[65] W. D. Fisher, T. K. Camp, and V. V. Krzhizhanovskaya, "Anomaly detection in Earth dam and levee passive seismic data using support vector machines and automatic feature selection," *J. Comput. Sci.*, vol. 20, pp. 143–153, May 2017.

[66] M. Flach *et al.*, "Multivariate anomaly detection for Earth observations: A comparison of algorithms and feature extraction techniques," *Earth Syst. Dyn.*, vol. 8, no. 3, pp. 677–696, Aug. 2017.

[67] Y. Wu, Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson, "DeepDetect: A cascaded region-based densely connected network for seismic event detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 62–75, Jan. 2019.

[68] T. Jiang, Y. Li, W. Xie, and Q. Du, "Discriminative reconstruction constrained generative adversarial network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4666–4679, Jul. 2020.

[69] T. I. Oprea, "Chemical space navigation in lead discovery," *Current Opinion Chem. Biol.*, vol. 6, no. 3, pp. 384–389, Jun. 2002.

[70] P. S. Gromski, A. B. Henson, J. M. Granda, and L. Cronin, "How to explore chemical space using algorithms and automation," *Nature Rev. Chem.*, vol. 3, no. 2, pp. 119–128, Feb. 2019.

[71] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings Bioinf.*, vol. 18, no. 5, pp. 851–869, 2017.

[72] S. A. Tomlins, "Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer," *Science*, vol. 310, no. 5748, pp. 644–648, Oct. 2005.

[73] R. Tibshirani and T. Hastie, "Outlier sums for differential gene expression analysis," *Biostatistics*, vol. 8, no. 1, pp. 2–8, Jan. 2007.

[74] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J.-R. Vlimant, "Variational autoencoders for new physics mining at the large hadron collider," *J. High Energy Phys.*, vol. 2019, no. 5, p. 36, May 2019.

[75] Y. A. Kharkov, V. E. Sotskov, A. A. Karazeev, E. O. Kiktenko, and A. K. Fedorov, "Revealing quantum chaos with machine learning," *Phys. Rev. B, Condens. Matter*, vol. 101, no. 6, Feb. 2020, Art. no. 064406.

[76] P. Protopapas, J. M. Giammarco, L. Faccioli, M. F. Struble, R. Dave, and C. Alcock, "Finding outlier light curves in catalogues of periodic variable stars," *Monthly Notices Roy. Astronomical Soc.*, vol. 369, no. 2, pp. 677–696, Jun. 2006.

[77] H. Dutta, C. Giannella, K. Borne, and H. Kargupta, "Distributed top-K outlier detection from astronomy catalogs using the DEMAC system," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2007, pp. 473–478.

[78] M. Henrion, D. J. Mortlock, D. J. Hand, and A. Gandy, "Classification and anomaly detection for astronomical survey data," in *Astrostatistical Challenges for the New Astronomy*. New York, NY, USA: Springer, 2013, pp. 149–184.

[79] E. Reyes and P. A. Estévez, "Transformation based deep anomaly detection in astronomical images," 2020, *arXiv:2005.07779*. [Online]. Available: http://arxiv.org/abs/2005.07779

[80] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[81] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[82] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[83] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[85] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[86] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[87] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[88] S. Ren, K. He, R. Girshick, and J. Sun, "Faster

R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[89] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[91] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[92] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4401–4410.

[93] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10687–10698.

[94] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1096–1104.

[95] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[96] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[97] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[98] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[99] A. Hannun *et al.*, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*. [Online]. Available: http://arxiv.org/abs/1412.5567

[100] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and mandarin," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 173–182.

[101] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 4960–4964.

[102] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2041–2053, Dec. 2019.

[103] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, Sep. 2019, pp. 3465–3469.

[104] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[105] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[106] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[107] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[108] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[109] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 427–431.

[110] M. Peters *et al.*, "Deep contextualized word representations," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 2227–2237.

[111] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.

[112] C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung, "Transferable multi-domain state generator for task-oriented dialogue systems," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 808–819.

[113] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[114] T. Lengauer, O. Sander, S. Sierra, A. Thielen, and R. Kaiser, "Bioinformatics prediction of HIV coreceptor usage," *Nature Biotechnol.*, vol. 25, no. 12, p. 1407, 2007.

[115] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Commun.*, vol. 5, no. 1, p. 4308, Sep. 2014.

[116] K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Commun.*, vol. 8, no. 1, pp. 1–8, Apr. 2017.

[117] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, Feb. 2017.

[118] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, "Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions," *Nature Commun.*, vol. 10, no. 1, p. 5024, Dec. 2019.

[119] P. Jurmeister *et al.*, "Machine learning analysis of DNA methylation profiles distinguishes primary lung squamous cell carcinomas from head and neck metastases," *Sci. Transl. Med.*, vol. 11, no. 509, Sep. 2019, Art. no. eaaw8513.

[120] F. Klauschen *et al.*, "Scoring of tumor-infiltrating lymphocytes: From visual estimation to machine learning," *Seminars Cancer Biol.*, vol. 52, no. 2, p. 151, 2018.

[121] F. Arcadu, F. Benmansour, A. Maunz, J. Willis, Z. Haskova, and M. Prunotto, "Deep learning algorithm predicts diabetic retinopathy progression in individual patients," *NPJ Digit. Med.*, vol. 2, no. 1, pp. 1–9, Dec. 2019.

[122] D. Ardila *et al.*, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," *Nature Med.*, vol. 25, no. 6, pp. 954–961, Jun. 2019.

[123] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nature Med.*, vol. 25, no. 1, pp. 24–29, 2019.

[124] K. Faust *et al.*, "Visualizing histopathologic deep learning classification and anomaly detection using nonlinear feature space dimensionality reduction," *BMC Bioinf.*, vol. 19, no. 1, p. 173, Dec. 2018.

[125] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 36–51.

[126] J. Chen, S. Sathe, C. C. Aggarwal, and D. S. Turaga, "Outlier detection with autoencoder ensembles," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 90–98.

[127] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 665–674.

[128] B. Zong *et al.*, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[129] C. Aytekin, X. Ni, F. Cricri, and E. Aksu, "Clustering and unsupervised anomaly detection with l_2 normalized deep auto-encoder representations," in *Proc. Int. Joint Conf. Neural Netw.*, 2018, pp. 1–6.

[130] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 481–490.

[131] C. Huang, F. Ye, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," 2019, *arXiv:1911.10676*. [Online]. Available: http://arxiv.org/abs/1911.10676

[132] D. Gong *et al.*, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1705–1714.

[133] P. Oza and V. M. Patel, "C2AE: Class conditioned auto-encoder for open-set recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2307–2316.

[134] D. T. Nguyen, Z. Lou, M. Klar, and T. Brox, "Anomaly detection with multiple-hypotheses predictions," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 4800–4809.

[135] K. H. Kim *et al.*, "RaPP: Novelty detection with reconstruction along projection pathway," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[136] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[137] L. Ruff *et al.*, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 4390–4399.

[138] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.

[139] P. Oza and V. M. Patel, "One-class convolutional neural network," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 277–281, Feb. 2019.

[140] 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 *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4061–4071.

[141] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: One-class novelty detection using GANs with constrained latent representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2898–2906.

[142] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5450–5463, Nov. 2019.

[143] J. Wang and A. Cherian, "GODS: Generalized one-class discriminative subspaces for anomaly detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 8201–8211.

[144] L. Ruff *et al.*, "Deep semi-supervised anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[145] Z. Ghafoori and C. Leckie, "Deep multi-sphere support vector data description," in *Proc. SIAM Int. Conf. Data Mining*, 2020, pp. 109–117.

[146] R. Chalapathy, E. Toth, and S. Chawla, "Group anomaly detection using deep generative models," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, 2018, pp. 173–189.

[147] L. Deecke, R. A. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Image anomaly detection with generative adversarial networks," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, 2018, pp. 3–17.

[148] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised

anomaly detection via adversarial training," in *Computer Vision—ACCV*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham, Switzerland: Springer, 2019, pp. 622–637.

[149] H. Choi, E. Jang, and A. A. Alemi, "WAIC, but why? Generative ensembles for robust anomaly detection," 2018, *arXiv:1810.01392*. [Online]. Available: http://arxiv.org/abs/1810.01392

[150] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6822–6833.

[151] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2018, pp. 727–736.

[152] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9758–9769.

[153] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15637–15648.

[154] S. Wang *et al.*, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5960–5973.

[155] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[156] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty detection via contrastive learning on distributionally shifted instances," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[157] M. Markou and S. Singh, "Novelty detection: A review—Part 1: Statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, Dec. 2003.

[158] M. Markou and S. Singh, "Novelty detection: A review—Part 2: Neural network based approaches," *Signal Process.*, vol. 83, no. 12, pp. 2499–2521, Dec. 2003.

[159] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[160] S. Walfish, "A review of statistical outlier methods," *Pharmaceutical Technol.*, vol. 30, no. 11, pp. 1–5, 2006.

[161] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[162] A. S. Hadi, R. Imon, and M. Werner, "Detection of outliers," *Wiley Interdiscipl. Rev. Comput. Statist.*, vol. 1, no. 1, pp. 57–70, 2009.

[163] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *Comput. J.*, vol. 54, no. 4, pp. 570–588, Apr. 2011.

[164] K. Singh and S. Upadhyaya, "Outlier detection: Applications and techniques," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, p. 307, 2012.

[165] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Stat. Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.

[166] H. Aguinis, R. K. Gottfredson, and H. Joo, "Best-practice recommendations for defining, identifying, and handling outliers," *Organizational Res. Methods*, vol. 16, no. 2, pp. 270–301, Apr. 2013.

[167] J. Zhang, "Advancements of outlier detection: A survey," *ICST Trans. Scalable Inf. Syst.*, vol. 13, no. 1, pp. 1–26, 2013.

[168] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, Jun. 2014.

[169] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[170] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Comput. Sci.*, vol. 60, pp. 708–713, 2015.

[171] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 626–688, May 2015.

[172] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *Wiley Interdiscipl. Rev. Comput. Statist.*, vol. 7, no. 3, pp. 223–247, May 2015.

[173] J. Tamboli and M. Shukla, "A survey of outlier detection algorithms for data streams," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop.*, 2016, pp. 3535–3540.

[174] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, Apr. 2016, Art. no. e0152173.

[175] X. Xu, H. Liu, and M. Yao, "Recent progress of anomaly detection," *Complexity*, vol. 2019, pp. 1–11, Jan. 2019.

[176] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 107964–108000, 2019.

[177] V. Barnett and T. Lewis, *Outliers in Statistical Data*, 3rd ed. Hoboken, NJ, USA: Wiley, 1994.

[178] P. J. Rousseeuw and A. M. Leroy, *Robust Regression Outlier Detection*. Hoboken, NJ, USA: Wiley, 2005.

[179] C. C. Aggarwal, *Outlier Analysis*, 2nd ed. Springer, 2017.

[180] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[181] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, "A survey on GANs for anomaly detection," 2019, *arXiv:1906.11632*. [Online]. Available: http://arxiv.org/abs/1906.11632

[182] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," 2020, *arXiv:2007.02500*. [Online]. Available: http://arxiv.org/abs/2007.02500

[183] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[184] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[185] S. García, J. Luengo, and F. Herrera, *Data Preprocessing Data Mining*, 1st ed. Cham, Switzerland: Springer, 2015.

[186] D. Rumsfeld, *Known Unknown: A Memoir*. Baltimore, MD, USA: Penguin, 2011.

[187] F. J. Anscombe, "Rejection of outliers," *Technometrics*, vol. 2, no. 2, pp. 123–146, May 1960.

[188] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, Feb. 1969.

[189] D. M. Hawkins, *Identification of Outliers*, vol. 11. Dordrecht, The Netherlands: Springer, 1980.

[190] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9592–9600.

[191] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 5, pp. 631–645, May 2007.

[192] K. Smets, B. Verdonk, and E. M. Jordaan, "Discovering novelty in spatio/temporal data using one-class support vector machines," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 2956–2963.

[193] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 823–839, May 2012.

[194] W. Lu *et al.*, "Unsupervised sequential outlier detection with deep architectures," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4321–4330, Sep. 2017.

[195] W. Samek, S. Nakajima, M. Kawanabe, and K.-R. Müller, "On robust parameter estimation in brain–computer interfacing," *J. Neural Eng.*, vol. 14, no. 6, Dec. 2017, Art. no. 061001.

[196] L. Xiong, B. Póczos, and J. G. Schneider, "Group anomaly detection using flexible genre models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1071–1079.

[197] K. Muandet and B. Schölkopf, "One-class support measure machines for group anomaly detection," in *Proc. Conf. Uncertainty Artif. Intell.*, 2013, pp. 449–458.

[198] R. Yu, X. He, and Y. Liu, "GLAD: Group anomaly detection in social media analysis," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 2, pp. 1–22, 2015.

[199] L. Bontemps *et al.*, "Collective anomaly detection based on long short-term memory recurrent neural networks," in *Proc. Int. Conf. Future Data Secur. Eng.* Springer, 2016, pp. 141–152.

[200] F. Ahmed and A. Courville, "Detecting semantic anomalies," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3154–3162.

[201] A. J. Fox, "Outliers in time series," *J. Roy. Stat. Soc. Ser. B, Methodol.*, vol. 34, no. 3, pp. 350–363, 1972.

[202] R. S. Tsay, "Outliers, level shifts, and variance changes in time series," *J. Forecasting*, vol. 7, no. 1, pp. 1–20, Jan. 1988.

[203] R. S. Tsay, D. Peña, and A. E. Pankratz, "Outliers in multivariate time series," *Biometrika*, vol. 87, no. 4, pp. 789–804, Dec. 2000.

[204] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms—The numenta anomaly benchmark," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2015, pp. 38–44.

[205] S. Chawla and P. Sun, "SLOM: A new measure for local spatial outliers," *Knowl. Inf. Syst.*, vol. 9, no. 4, pp. 412–429, Apr. 2006.

[206] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, 2014.

[207] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 631–636.

[208] J. Höner, S. Nakajima, A. Bauer, K.-R. Müller, and N. Görnitz, "Minimizing trust leaks for robust sybil detection," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1520–1528.

[209] M. Ahmed, "Collective anomaly detection techniques for network traffic analysis," *Ann. Data Sci.*, vol. 5, no. 4, pp. 497–512, Dec. 2018.

[210] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 4114–4124.

[211] B. Schölkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA, USA: MIT Press, 2002.

[212] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *J. Mach. Learn. Res.*, vol. 6, pp. 211–232, Feb. 2005.

[213] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[214] W. Polonik, "Measuring mass concentrations and estimating density contour clusters—An excess mass approach," *Ann. Statist.*, vol. 23, no. 3, pp. 855–881, Jun. 1995.

[215] A. B. Tsybakov, "On nonparametric estimation of density level sets," *Ann. Statist.*, vol. 25, no. 3, pp. 948–969, Jun. 1997.

[216] S. Ben-David and M. Lindenbaum, "Learning distributions by their density levels: A paradigm for learning without a teacher," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 171–182, Aug. 1997.

[217] P. Rigollet and R. Vert, "Optimal rates for plug-in estimators of density level sets," *Bernoulli*, vol. 15, no. 4, pp. 1154–1178, Nov. 2009.

[218] W. Polonik, "Minimum vol. sets, and generalized quantile processes," *Stochastic Processes Their Appl.*, vol. 69, no. 1, pp. 1–24, 1997.

[219] J. N. Garcia, Z. Kutalik, K.-H. Cho, and O. Wolkenhauer, "Level sets and minimum volume sets of probability density functions," *Int.*

J. Approx. Reasoning, vol. 34, no. 1, pp. 25–47, 2003.

[220] C. D. Scott and R. D. Nowak, "Learning minimum volume sets," *J. Mach. Learn. Res.*, vol. 7, pp. 665–704, Apr. 2006.

[221] L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet, "Robust novelty detection with single-class MPM," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 929–936.

[222] A. K. Menon and R. C. Williamson, "A loss framework for calibrated anomaly detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1494–1504.

[223] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, nos. 11–13, pp. 1191–1199, Nov. 1999.

[224] D. M. J. Tax, "One-class classification," Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, 2001. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3Ae588fc3e-7503-4013-9b6a-73c7b7f6b173

[225] S. Clémençon and J. Jakubowicz, "Scoring anomalies: A M-estimation formulation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2013, pp. 659–667.

[226] N. Goix, A. Sabourin, and S. Clémençon, "On anomaly ranking and excess-mass curves," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 287–295.

[227] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based Influence Functions*. Hoboken, NJ, USA: Wiley, 2005.

[228] Y. Liu and Y. F. Zheng, "Minimum enclosing and maximum excluding machine for pattern description and discrimination," in *Proc. Int. Conf. Pattern Recognit.*, 2006, pp. 129–132.

[229] N. Goernitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *J. Artif. Intell. Res.*, vol. 46, pp. 235–262, Feb. 2013.

[230] E. Min, J. Long, Q. Liu, J. Cui, Z. Cai, and J. Ma, "SU-IDS: A semi-supervised and unsupervised framework for network intrusion detection," in *Proc. Int. Conf. Cloud Comput. Secur.*, 2018, pp. 322–334.

[231] B. Kiran, D. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *J. Imag.*, vol. 4, no. 2, p. 36, Feb. 2018.

[232] M. A. Siddiqui, A. Fern, T. G. Dietterich, R. Wright, A. Theriault, and D. W. Archer, "Feedback-guided anomaly discovery via online optimization," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2200–2209.

[233] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[234] F. Denis, "PAC learning from positive statistical queries," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 1998, pp. 112–126.

[235] B. Zhang and W. Zuo, "Learning from positive and unlabeled examples: A survey," in *Proc. IEEE Int. Symp. Inf. Process.*, May 2008, pp. 650–654.

[236] M. C. Du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 703–711.

[237] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls, "Semi-supervised one-class support vector machines for classification of remote sensing Sata," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 8, pp. 3188–3197, May 2010.

[238] G. Blanchard, G. Lee, and C. Scott, "Semi-supervised novelty detection," *J. Mach. Learn. Res.*, vol. 11, pp. 2973–3009, Nov. 2010.

[239] H. Song, Z. Jiang, A. Men, and B. Yang, "A hybrid semi-supervised anomaly detection model for high-dimensional data," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–9, Nov. 2017.

[240] S. Liu, R. Garrepalli, T. Dietterich, A. Fern, and D. Hendrycks, "Open category detection with PAC guarantees," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 3169–3178.

[241] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[242] T. Che *et al.*, "Deep verifier networks: Verification of deep discriminative models with deep generative models," 2019, *arXiv:1911.07421*. [Online]. Available: http://arxiv.org/abs/1911.07421

[243] R. T. Schirrmeister, Y. Zhou, T. Ball, and D. Zhang, "Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[244] G. Boracchi, D. Carrera, C. Cervellera, and D. Maccio, "QuantTree: Histograms for change detection in multivariate data streams," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 639–648.

[245] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, no. May, pp. 985–1005, 2007.

[246] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[247] M. Sugiyama and M. Kawanabe, *Mach. Learn. Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. Cambridge, MA, USA: MIT Press, 2012.

[248] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, Jun. 2010.

[249] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018.

[250] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking Clever Hans predictors and assessing what machines really learn," *Nature Commun.*, vol. 10, no. 1, p. 1096, Dec. 2019.

[251] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Lecture Notes in Computer Science), vol. 11700. Springer, 2019.

[252] W. Härdle, *Applied Nonparametric Regression*, no. 19. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[253] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, "Informal identification of outliers in medical data," in *Proc. 5th Int. Workshop Intell. Data Anal. Med. Pharmacol.*, vol. 1, 2000, pp. 20–24.

[254] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.

[255] S. Roberts and L. Tarassenko, "A probabilistic resource allocating network for novelty detection," *Neural Comput.*, vol. 6, no. 2, pp. 270–284, Mar. 1994.

[256] C. M. Bishop, "Novelty detection and neural network validation," *IEE Proc.-Vis., Image Signal Process.*, vol. 141, no. 4, pp. 217–222, 1994.

[257] L. Devroye and L. Györfi, *Nonparametric Density Estimation: The L1 View*. New York, NY, USA: Wiley, 1985.

[258] S. Frühwirth-Schnatter, *Finite Mixture Markov Switching Models*. New York, NY, USA: Springer, 2006.

[259] J. Kim and C. D. Scott, "Robust kernel density estimation," *J. Mach. Learn. Res.*, vol. 13, no. 82, pp. 2529–2565, 2012.

[260] R. Vandermeulen and C. Scott, "Consistency of robust kernel density estimators," in *Proc. Conf. Learn. Theory*, 2013, pp. 568–591.

[261] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," in *Proc. 5th Int. Conf. Ind. Eng. Appl. (ICIEA)*, Apr. 2018, pp. 355–361.

[262] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski, "Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines," in *Proc. AAAI Conf. Artif. Intell.*, 1983, pp. 109–113.

[263] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.

[264] Y. Lecun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, *A Tutorial on Energy-Based Learning*. Cambridge, MA, USA: MIT Press, 2006.

[265] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[266] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 681–688.

[267] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[268] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[269] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 448–455.

[270] J. Ngiam, Z. Chen, P. W. Koh, and A. Ng, "Learning deep energy models," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1105–1112.

[271] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, vol. 48, 2016, pp. 1100–1109.

[272] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[273] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, vol. 32, 2014, pp. 1278–1286.

[274] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[275] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[276] H. Xu *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in Web applications," in *Proc. World Wide Web Conf.*, 2018, pp. 187–196.

[277] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" in *Proc. Int. Conf. Learn. Represent.*, 2019.

[278] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lect. IE*, vol. 2, no. 1, pp. 1–18, 2015.

[279] T. Salimans *et al.*, "Improved techniques for training gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[280] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 214–223.

[281] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[282] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[283] G. Papamakarios, E. Nalisnick, D. Jimenez Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," 2019, *arXiv:1912.02762*. [Online]. Available: http://arxiv.org/abs/1912.02762

[284] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 7, 2020, doi: 10.1109/TPAMI.2020.2992934.

[285] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[286] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural autoregressive flows," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 2078–2087.

[287] F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," *Science*, vol. 365, no. 6457, Sep. 2019, Art. no. eaaw1147.

[288] B. Nachman and D. Shih, "Anomaly detection with density estimation," *Phys. Rev. D, Part. Fields*, vol. 101, no. 7, Apr. 2020, Art. no. 075042.

[289] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multi-modal anomaly detection," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1326–1333, Apr. 2020.

[290] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Why normalizing flows fail to detect out-of-distribution data," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[291] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[292] S. Suh, D. H. Chae, H.-G. Kang, and S. Choi, "Echo-state conditional variational autoencoder for anomaly detection," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2016, pp. 1015–1022.

[293] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, "Noise flow: Noise modeling with conditional normalizing flows," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3165–3173.

[294] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.*, vol. 2019, pp. 703–716.

[295] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 10–21.

[296] L. Chen *et al.*, "Adversarial text generation via feature-mover's distance," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4666–4677.

[297] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 2323–2332.

[298] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "NetGAN: Generating graphs via random walks," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 610–619.

[299] R. Liao, Y. Li, Y. Song, S. Wang, W. Hamilton, D. K. Duvenaud, R. Urtasun, and R. Zemel, "Efficient graph generation with graph recurrent attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4255–4265.

[300] V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.

[301] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," in *Proc. World Congr. Neural Netw.*, 1993, pp. 797–801.

[302] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Netw.*, vol. 9, no. 3, pp. 463–474, 1996.

[303] S. S. Khan and M. G. Madden, "One-class classification: Taxonomy of study and review of techniques," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 345–374, Jun. 2014.

[304] T. Minter, "Single-class classification," in *Proc. LARS Symposia*, 1975, p. 54.

[305] R. El-Yaniv and M. Nisenson, "Optimal single-class classification strategies," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 377–384.

[306] P. J. Rousseeuw, "Multivariate estimation with high breakdown point," *Math. Statist. Appl.*, vol. 8, pp. 283–297, 1985.

[307] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, Aug. 1999.

[308] A. Muñoz and J. M. Moguerza, "Estimation of high-density regions using one-class neighbor machines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 476–480, Mar. 2006.

[309] B. Schölkopf *et al.*, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1000–1017, Sep. 1999.

[310] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, no. Dec., pp. 139–154, 2001.

[311] R. Vert and J.-P. Vert, "Consistency and convergence rates of one-class SVMs and related algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 817–854, May 2006.

[312] G. Lee and C. D. Scott, "The one class support vector machine solution path," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 2007, pp. 521–524.

[313] K. Sjöstrand and R. Larsen, "The entire regularization path for the support vector domain description," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent*, 2006, pp. 241–248.

[314] G. Lee and C. Scott, "Nested support vector machines," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1648–1660, Mar. 2010.

[315] A. Glazer, M. Lindenbaum, and S. Markovitch, "q-OCSVM: A q-quantile estimator for high-dimensional distributions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 503–511.

[316] N. Görnitz, L. A. Lima, K.-R. Müller, M. Kloft, and S. Nakajima, "Support vector data descriptions and k-means clustering: One class?" *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 3994–4006, Sep. 2018.

[317] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 47–56.

[318] C. Gautam, R. Balaji, K. Sudharsan, A. Tiwari, and K. Ahuja, "Localized multiple kernel learning for anomaly detection: One-class classification," *Knowl.-Based Syst.*, vol. 165, pp. 241–252, Feb. 2019.

[319] G. Ratsch, S. Mika, B. Scholkopf, and K.-R. Muller, "Constructing boosting algorithms from SVMs: An application to one-class classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1184–1199, Sep. 2002.

[320] V. Roth, "Outlier detection with one-class kernel Fisher discriminants," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1169–1176.

[321] V. Roth, "Kernel Fisher discriminants for outlier detection," *Neural Comput.*, vol. 18, no. 4, pp. 942–960, Apr. 2006.

[322] F. Dufrenois, "A one-class kernel Fisher criterion for outlier detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 982–994, May 2015.

[323] A. Ghasemi, H. R. Rabiee, M. T. Manzuri, and M. H. Rohban, "A Bayesian approach to the data description problem," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 907–913.

[324] M. Stolpe, K. Bhaduri, K. Das, and K. Morik, "Anomaly detection in vertically partitioned data by distributed Core vector machines," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 321–336.

[325] H. Jiang, H. Wang, W. Hu, D. Kakde, and A. Chaudhuri, "Fast incremental SVDD learning algorithm with the Gaussian kernel," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3991–3998.

[326] W. Liu, G. Hua, and J. R. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3826–3833.

[327] P. Wu, J. Liu, and F. Shen, "A deep one-class neural network for anomalous event detection in complex scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2609–2622, Jul. 2020.

[328] R. Chalapathy, A. Krishna Menon, and S. Chawla, "Anomaly detection using one-class neural networks," 2018, *arXiv:1802.06360*. [Online]. Available: http://arxiv.org/abs/1802.06360

[329] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks Trade* (Lecture Notes in Computer Science), vol. 7700, G. Montavon and G. B. Orr, Eds. Berlin, Germany: Springer, 2012, pp. 9–48.

[330] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[331] G. Goh, "Why momentum really works," *Distill*, 2017. [Online]. Available: http://distill.pub/2017/momentum, doi: 10.23915/distill.00006.

[332] L. Ruff, R. A. Vandermeulen, B. Joe Franks, K.-R. Müller, and M. Kloft, "Rethinking assumptions in deep anomaly detection," 2020, *arXiv:2006.00339*. [Online]. Available: http://arxiv.org/abs/2006.00339

[333] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain, "DROCC: Deep robust one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11335–11345.

[334] J. Kauffmann, L. Ruff, G. Montavon, and K.-R. Müller, "The clever Hans effect in anomaly detection," 2020, *arXiv:2006.10609*. [Online]. Available: http://arxiv.org/abs/2006.10609

[335] P. Chong, L. Ruff, M. Kloft, and A. Binder, "Simple and effective prevention of mode collapse in deep one-class classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–9.

[336] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 353–362.

[337] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller, "Explainable deep one-class classification," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[338] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[339] M. Sabokrou, M. Fathy, G. Zhao, and E. Adeli, "Deep end-to-end one-class classifier," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, 2020, doi: 10.1109/TNNLS.2020.2979049.

[340] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[341] G. Steinbuss and K. Böhm, "Generating artificial outliers in the absence of genuine ones—A survey," 2020, *arXiv:2006.03646*. [Online]. Available: http://arxiv.org/abs/2006.03646

[342] J. P. Theiler and D. M. Cai, "Resampling approach for anomaly detection in multispectral images," *Proc. SPIE*, vol. 5093, pp. 230–240, Sep. 2003.

[343] M. A. Davenport, R. G. Baraniuk, and C. D. Scott, "Learning minimum volume sets with support vector machines," in *Proc. IEEE Signal Process. Soc. Workshop Mach. Learn. Signal Process.*, Sep. 2006, pp. 301–306.

[344] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowl. Inf. Syst.*, vol. 6, no. 5, pp. 507–527, Sep. 2004.

[345] P. Cheema *et al.*, "On structural health monitoring using tensor analysis and support vector machine with artificial negative data," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1813–1822.

[346] N. Abe, B. Zadrozny, and J. Langford, "Outlier detection by active learning," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 504–509.

[347] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman, "ALADIN: Active learning of anomalies to detect intrusions," Microsoft Res., New York, NY, USA, Tech. Rep. MSR-TR-2008-24, 2008.

[348] N. Görnitz, M. Kloft, and U. Brefeld, "Active and semi-supervised data domain description," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, 2009, pp. 407–422.

[349] D. Pelleg and A. W. Moore, "Active learning for anomaly and rare-category detection," in *Proc.*

Adv. Neural Inf. Process. Syst., 2005, pp. 1073–1080.

[350] M. Du, Z. Chen, C. Liu, R. Oak, and D. Song, "Lifelong anomaly detection through unlearning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 1283–1297.

[351] N. Japkowicz, C. Myers, and M. Gluck, "A novelty detection approach to classification," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 1, 1995, pp. 518–523.

[352] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*, vol. 2454, 2002, pp. 170–180.

[353] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. New York, NY, USA: Springer, 2007.

[354] R. Pless and R. Souvenir, "A survey of manifold learning for images," *IPSJ Trans. Comput. Vis. Appl.*, vol. 1, pp. 83–94, 2009.

[355] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[356] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Berlin, Germany: Springer, 2001.

[357] L. van der Maaten, E. Postma, and J. van den Herik, "Dimensionality reduction: A comparative review," Tilburg Centre Creative Comput., Tilburg Univ., Tilburg, The Netherlands, Tech. Rep. TiCC-TR 2009-005, 2009.

[358] J. Schmidhuber, "Learning factorial codes by predictability minimization," *Neural Comput.*, vol. 4, no. 6, pp. 863–879, Nov. 1992.

[359] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," in *Proc. Workshop Bayesian Deep Learn. (NeurIPS)*, 2018.

[360] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[361] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression* (The Springer International Series in Engineering and Computer Science), vol. 159. Boston, MA, USA: Springer, 1992.

[362] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999.

[363] C. M. Bishop, "Bayesian PCA," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 382–388.

[364] I. T. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics), 2nd ed. New York, NY, USA: Springer, 2002.

[365] D. M. Hawkins, "The detection of errors in multivariate data using principal components," *J. Amer. Stat. Assoc.*, vol. 69, no. 346, pp. 340–344, Jun. 1974.

[366] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, Aug. 1979.

[367] L. Parra, G. Deco, and S. Miesbach, "Statistical independence and novelty detection with information preserving nonlinear maps," *Neural Comput.*, vol. 8, no. 2, pp. 260–269, Feb. 1996.

[368] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. IEEE Int. Conf. Data Mining*, Jan. 2003, pp. 353–365.

[369] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft, "In-network PCA and anomaly detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 617–624.

[370] V. Sharan, P. Gopalan, and U. Wieder, "Efficient anomaly detection via matrix sketching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8069–8080.

[371] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 47.

[372] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, Sep. 2008.

[373] M. H. Nguyen and F. Torre, "Robust kernel principal component analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1185–1192.

[374] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[375] Y. Xiao, H. Wang, W. Xu, and J. Zhou, "L1 norm based KPCA for novelty detection," *Pattern Recognit.*, vol. 46, no. 1, pp. 389–396, Jan. 2013.

[376] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, no. 3, pp. 267–273, 1982.

[377] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing Explorations in the Microstructure of Cognition*. Cambridge, MA, USA: MIT Press, 1986, ch. 8, pp. 318–362.

[378] D. H. Ballard, "Modular learning in neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 1987, pp. 279–284.

[379] G. E. Hinton, "Connectionist learning procedures," *Artif. Intell.*, vol. 40, nos. 1–3, pp. 185–234, Sep. 1989.

[380] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.

[381] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[382] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, no. 1, pp. 53–58, Jan. 1989.

[383] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Netw.*, vol. 5, no. 6, pp. 927–935, Nov. 1992.

[384] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996.

[385] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.

[386] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, vol. 12, no. 2, pp. 337–365, Feb. 2000.

[387] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 801–808.

[388] A. Makhzani and B. Frey, "k-sparse autoencoders," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.

[389] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643–649, Jan. 2018.

[390] D. Arpit, Y. Zhou, H. Ngo, and V. Govindaraju, "Why regularized auto-encoders learn sparse representation?" in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 136–144.

[391] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[392] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[393] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 833–840.

[394] S. You, K. C. Tezcan, X. Chen, and E. Konukoglu, "Unsupervised lesion detection via image restoration with a normative prior," in *Proc. Int. Conf. Med. Imag. Deep Learn.*, 2019, pp. 540–556.

[395] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[396] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: http://arxiv.org/abs/1511.05644

[397] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*. [Online]. Available: http://arxiv.org/abs/1607.00148

[398] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2725–2732.

[399] G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated gradient representations for anomaly detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 206–226.

[400] C. D. Hofer, R. Kwitt, M. Dixit, and M. Niethammer, "Connectivity-optimized representation learning via persistent homology," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 2751–2760.

[401] T. Amarbayasgalan, B. Jargalsaikhan, and K. Ryu, "Unsupervised novelty detection using deep autoencoders with density based clustering," *Appl. Sci.*, vol. 8, no. 9, p. 1468, Aug. 2018.

[402] N. Sarafijanovic-Djukic and J. Davis, "Fast distance-based anomaly detection in images using an inception-like autoencoder," in *Proc. Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2019, pp. 493–508.

[403] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[404] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites," *J. für die Reine und Angewandte Mathematik*, vol. 1908, no. 133, pp. 97–178, 1908.

[405] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs," *J. für die Reine und Angewandte Mathematik*, vol. 1908, no. 134, pp. 198–287, 1908.

[406] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means, spectral clustering and normalized cuts," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 551–556.

[407] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 478–487.

[408] A. Van Den Oord and O. Vinyals, "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6306–6315.

[409] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14866–14876.

[410] M. Kampffmeyer, S. Løkse, F. M. Bianchi, L. Livi, A.-B. Salberg, and R. Jenssen, "Deep divergence-based approach to clustering," *Neural Netw.*, vol. 113, pp. 91–101, May 2019

[411] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3861–3870.

[412] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[413] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 517–526.

[414] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[415] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[416] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, 2nd ed. New York, NY, USA: Academic, 2020.

[417] D. J. C. MacKay, "A practical Bayesian framework for backpropagation networks," *Neural Comput.*,

vol. 4, no. 3, pp. 448–472, May 1992.

[418] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 1613–1622.

[419] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, and M. Kloft, "Deep support vector data description for unsupervised and semi-supervised anomaly detection," in *Proc. ICML Workshop Uncertainty Robustness Deep Learn.*, 2019, pp. 9–15.

[420] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, and K.-R. Müller, "From outliers to prototypes: Ordering data," *Neurocomputing*, vol. 69, nos. 13–15, pp. 1608–1618, Aug. 2006.

[421] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2250–2258.

[422] X. Gu, L. Akoglu, and A. Rinaldo, "Statistical analysis of nearest neighbor methods for anomaly detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10923–10933.

[423] P. Juszczak, D. M. J. Tax, E. Pe ¸kalska, and R. P. W. Duin, "Minimum spanning tree based one-class classifier," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1859–1869, Mar. 2009.

[424] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[425] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 2712–2721.

[426] L. Bergman, N. Cohen, and Y. Hoshen, "Deep nearest neighbor anomaly detection," 2020, *arXiv:2002.10445*. [Online]. Available: http://arxiv.org/abs/2002.10445

[427] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *Proc. IEEE Secur. Privacy Workshops*, May 2013, pp. 98–104.

[428] N. Mu and J. Gilmer, "MNIST-C: A robustness benchmark for computer vision," 2019, *arXiv:1906.02337*. [Online]. Available: http://arxiv.org/abs/1906.02337

[429] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[430] P. Laskov, C. Schäfer, I. Kotenko, and K.-R. Müller, "Intrusion detection in unlabeled data with quarter-sphere support vector machines," *PIK-Praxis der Informationsverarbeitung und Kommunikation*, vol. 27, no. 4, pp. 228–236, Dec. 2004.

[431] M. Kloft and P. Laskov, "Security analysis of online centroid anomaly detection," *J. Mach. Learn. Res.*, vol. 13, no. 118, pp. 3681–3724, 2012.

[432] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "Systematic construction of anomaly detection benchmarks from real data," in *Proc. Workshop Outlier Detection Description (KDD)*, 2013, pp. 16–21.

[433] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "A meta-analysis of the anomaly detection problem," no. v2, pp. 1–35, 2016, *arXiv:1503.01158*. [Online]. Available: http://arxiv.org/abs/1503.01158

[434] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," 2019, *arXiv:1907.07174*. [Online]. Available: http://arxiv.org/abs/1907.07174

[435] W. Huang and P. Wei, "A PCB dataset for defects detection and classification," 2019, *arXiv:1901.08204*. [Online]. Available: http://arxiv.org/abs/1901.08204

[436] B. E. Bejnordi *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *J. Amer. Med. Assoc.*, vol. 318, no. 22, pp. 2199–2210, 2017.

[437] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2097–2106.

[438] D. Zimmerer *et al.*, *Medical Out-of-Distribution Analysis Challenge*. Mar. 2020. [Online]. Available: https://www.synapse.org/#!Synapse:syn21343101/wiki/599515, doi: 10.5281/zenodo.3784230.

[439] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.

[440] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 681–688.

[441] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.

[442] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017.

[443] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1939–1947.

[444] G. O. Campos *et al.*, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 891–927, Jul. 2016.

[445] S. Rayana. (2016). *ODDS Library*. [Online]. Available: http://odds.cs.stonybrook.edu

[446] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, Feb. 2018.

[447] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[448] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[449] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[450] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, Jul. 2014.

[451] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[452] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: Point estimates and confidence intervals," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, 2013, pp. 451–466.

[453] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[454] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 618–626.

[455] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3319–3328.

[456] Z. Qi, S. Khorram, and F. Li, "Visualizing deep networks by optimizing with integrated gradients," in *Proc. CVPR Workshops*, vol. 2, 2019, pp. 11890–11898.

[457] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[458] B. Micenková, R. T. Ng, X.-H. Dang, and I. Assent, "Explaining outliers by subspace separability," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 518–527.

[459] M. A. Siddiqui, A. Fern, T. G. Dietterich, and W.-K. Wong, "Sequential feature explanations for anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 1, pp. 1–22, Jan. 2019.

[460] J. Kauffmann, K.-R. Müller, and G. Montavon, "Towards explaining anomalies: A deep Taylor decomposition of one-class models," *Pattern Recognit.*, vol. 101, May 2020, Art. no. 107198.

[461] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[462] J. Kauffmann, M. Esders, G. Montavon, W. Samek, and K.-R. Müller, "From clustering to cluster explanations via neural networks," 2019, *arXiv:1906.07633*. [Online]. Available: http://arxiv.org/abs/1906.07633

[463] P. T. Huynh, A. M. Jarolimek, and S. Daye, "The false-negative mammogram," *Radiographics*, vol. 18, no. 5, pp. 1137–1154, 1998.

[464] M. Petticrew, A. Sowden, D. Lister-Sharp, and K. Wright, "False-negative results in screening programmes: Systematic review of impact and implications," *Health Technol. Assess.*, vol. 4, no. 5, pp. 1–120, 2000.

[465] M. S. Pepe, *The Statistical Evaluation of Medical Tests for Classification and Prediction*. London, U.K.: Oxford Univ. Press, 2003.

[466] X.-H. Zhou, N. A. Obuchowski, and D. K. McClish, *Statistical Methods in Diagnostic Medicine*, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.

[467] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Hoboken, NJ, USA: Wiley, 1973.

[468] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. New York, NY, USA: Academic, 2009.

[469] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 157–166.

[470] H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan, "Mining outliers with ensemble of heterogeneous detectors on random subspaces," in *DASFAA (1)* (Lecture Notes in Computer Science), vol. 5981. Berlin, Germany: Springer, 2010, pp. 368–383.

[471] M. L. Braun, J. M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *J. Mach. Learn. Res.*, vol. 9, pp. 1875–1908, Aug. 2008.

[472] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1050–1059.

[473] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6402–6413.

[474] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[475] Y. Ovadia *et al.*, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13991–14002.

[476] K. Bykov, M. M.-C. Höhne, K.-R. Müller, S. Nakajima, and M. Kloft, "How much can i trust you?—Quantifying uncertainties in explaining neural networks," 2020, *arXiv:2006.09000*. [Online]. Available: http://arxiv.org/abs/2006.09000

[477] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowl. Inf. Syst.*, vol. 26, no. 2, pp. 309–336, Feb. 2011.

[478] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.

[479] R. A. Vandermeulen, R. Saitenmacher, and A. Ritchie, "A proposal for supervised density

estimation," in *Proc. NeurIPS Pre-Registration Workshop*, 2020.

[480] C. K. Chow, "An optimum character recognition system using decision functions," *IRE Trans. Electron. Comput.*, vol. EC-6, no. 4, pp. 247–254, Dec. 1957.

[481] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Inf. Theory*, vol. 16, no. 1, pp. 41–46, Jan. 1970.

[482] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *J. Mach. Learn. Res.*, vol. 9, no. Aug, pp. 1823–1840, 2008.

[483] D. M. J. Tax and R. P. W. Duin, "Growing a multi-class classifier with a reject option," *Pattern Recognit. Lett.*, vol. 29, no. 10, pp. 1565–1570, Jul. 2008.

[484] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu, "Support vector machines with a reject option," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 537–544.

[485] C. Cortes, G. DeSalvo, and M. Mohri, "Learning with rejection," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2016, pp. 67–82.

[486] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4878–4887.

[487] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[488] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1321–1330.

[489] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," 2018, *arXiv:1802.04865*. [Online]. Available: http://arxiv.org/abs/1802.04865

[490] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[491] C. Ni, N. Charoenphakdee, J. Honda, and M. Sugiyama, "On the calibration of multiclass classification with rejection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2586–2596.

[492] A. Meinke and M. Hein, "Towards neural networks that provably know when they don't know," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[493] D. J. C. MacKay and M. N. Gibbs, "Density networks," in *Statistics and Neural Networks: Advances at the Interface*. USA: Oxford Univ. Press, 1998, pp. 129-146.

[494] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[495] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7167–7177.

[496] S. Choi and S.-Y. Chung, "Novelty detection via blurring," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[497] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[498] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.

[499] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1563–1572.

[500] L. Shu, H. Xu, and B. Liu, "DOC: Deep open classification of text documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2911–2916.

[501] H. Zhang, A. Li, J. Guo, and Y. Guo, "Hybrid models for open set recognition," 2020, *arXiv:2003.12506*. [Online]. Available: http://arxiv.org/abs/2003.12506

[502] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.

[503] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[504] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 39–57.

[505] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[506] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018.

[507] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 274–283.

[508] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[509] N. Carlini *et al.*, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*. [Online]. Available: http://arxiv.org/abs/1902.06705

[510] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 7472–7482.

[511] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 125–136.

[512] G. Shalev, Y. Adi, and J. Keshet, "Out-of-distribution detection using multiple semantic label representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7375–7385.

[513] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 2712–2721.

[514] A. R. Dhamija, M. Günther, and T. Boult, "Reducing network agnostophobia," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9157–9168.

[515] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13589–13600.

[516] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1721–1730.

[517] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Toward interpretable machine learning: Transparent deep neural networks and beyond," 2020, *arXiv:2003.07631*. [Online]. Available: http://arxiv.org/abs/2003.07631

[518] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Hum. Factors*, vol. 46, no. 1, pp. 50–80, 2004.

[519] H. Jiang, B. Kim, M. Guan, and M. Gupta, "To trust or not to trust a classifier," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5541–5552.

[520] Z. C. Lipton, "The doctor just won't accept that!" in *Proc. NIPS Interpretable ML Symp.*, 2017.

[521] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a 'right to explanation,'" *AI Mag.*, vol. 38, no. 3, pp. 50–57, 2017.

[522] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," 2016, *arXiv:1606.06565*. [Online]. Available: http://arxiv.org/abs/1606.06565

[523] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Proc. Robot. Sci. Syst.*, 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p64.html

[524] X. H. Dang, B. Micenková, I. Assent, and R. T. Ng, "Local outlier detection with interpretation," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, 2013, pp. 304–320.

[525] X. Hong Dang, I. Assent, R. T. Ng, A. Zimek, and E. Schubert, "Discriminative features for identifying and interpreting outliers," in *Proc. Int. Conf. Data Eng.*, Mar. 2014, pp. 88–99.

[526] L. Duan, G. Tang, J. Pei, J. Bailey, A. Campbell, and C. Tang, "Mining outlying aspects on numeric data," *Data Mining Knowl. Discovery*, vol. 29, no. 5, pp. 1116–1151, Sep. 2015.

[527] N. X. Vinh *et al.*, "Discovering outlying aspects in large datasets," *Data Mining Knowl. Discovery*, vol. 30, no. 6, pp. 1520–1555, Nov. 2016.

[528] M. Macha and L. Akoglu, "Explaining anomalies in groups with characterizing subspace rules," *Data Mining Knowl. Discovery*, vol. 32, no. 5, pp. 1444–1480, Sep. 2018.

[529] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[530] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[531] J. Wang, S. Sun, and Y. Yu, "Multivariate triangular quantile maps for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5061–5072.

[532] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14680–14691.

[533] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, "Input complexity and out-of-distribution detection with likelihood-based generative models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[534] G. E. Box, "Science and statistics," *J. Amer. Stat. Assoc.*, vol. 71, no. 356, pp. 791–799, 1976.

[535] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *Proc. VLDB Endowment*, vol. 11, no. 3, pp. 269–282, 2017.

[536] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, Jan. 2018.

[537] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data—AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 8, 2019, doi: 10.1109/TKDE.2019.2946162.

[538] T. Daniel, T. Kurutach, and A. Tamar, "Deep variational semi-supervised novelty detection," 2019, *arXiv:1911.04971*. [Online]. Available: http://arxiv.org/abs/1911.04971

[539] S. Das, W.-K. Wong, T. Dietterich, A. Fern, and A. Emmott, "Discovering anomalies by incorporating feedback from an expert," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 4, pp. 1–32, Jul. 2020.

[540] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, and O. Kao, "Self-attentive classification-based anomaly detection in unstructured logs," 2020, *arXiv:2008.09340*. [Online]. Available: http://arxiv.org/abs/2008.09340

[541] K. Ouardini *et al.*, "Towards practical unsupervised anomaly detection on retinal images," in *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. Cham, Switzerland: Springer, 2019, pp. 225–234.

[542] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole, "Weakly supervised disentanglement with guarantees," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[543] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen, "Weakly-supervised disentanglement without compromises," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7753–7764.

[544] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[545] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10709–10719.

[546] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.

[547] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1422–1430.

[548] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[549] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[550] J. Winkens *et al.*, "Contrastive training for improved out-of-distribution detection," 2020, *arXiv:2007.05566*. [Online]. Available: http://arxiv.org/abs/2007.05566

[551] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "A critical analysis of self-supervision, or what we can learn from a single image," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[552] E. Nalisnick, A. Matsukawa, Y. W. Teh, and B. Lakshminarayanan, "Detecting out-of-distribution inputs to deep generative models using typicality," in *Proc. Workshop Bayesian Deep Learn. (NeurIPS)*, 2019.

[553] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*, vol. 47. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[554] A. Tong, G. Wolf, and S. Krishnaswamyt, "Fixing bias in reconstruction-based anomaly detection with Lipschitz discriminators," in *Proc. IEEE 30th Int. Workshop Mach. Learn. for Signal Process. (MLSP)*, Sep. 2020, pp. 1–6.

[555] D. Krueger *et al.*, "Out-of-distribution generalization via risk extrapolation (REx)," 2020, *arXiv:2003.00688*. [Online]. Available: http://arxiv.org/abs/2003.00688

[556] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, July, Oct. 1948.

[557] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, Mar. 1988.

[558] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995.

[559] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[560] T. Berger, "Rate-distortion theory," in *Wiley Encyclopedia of Telecommunications*. New York, NY, USA: Wiley, 2003.

[561] I. Higgins *et al.*, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[562] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy, "Fixing a broken ELBO," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 159–168.

[563] S. Park, G. Adosoglou, and P. M. Pardalos, "Interpreting rate-distortion of variational autoencoder and using model uncertainty for anomaly detection," 2020, *arXiv:2005.01889*. [Online]. Available: http://arxiv.org/abs/2005.01889

[564] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proc. IEEE Symp. Secur. Privacy*, 2001, pp. 130–143.

[565] A. Høst-Madsen, E. Sabeti, and C. Walton, "Data discovery and anomaly detection using atypicality: Theory," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5302–5322, Sep. 2019.

[566] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[567] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[568] S. Sonnenburg *et al.*, "The need for open source software in machine learning," *J. Mach. Learn. Res.*, vol. 8, pp. 2443–2466, Oct. 2007.

ABOUT THE AUTHORS

Lukas Ruff received the bachelor's degree in mathematical finance from the University of Konstanz, Konstanz, Germany, in 2015, and the joint master's degree in statistics from the Humboldt University of Berlin (HU Berlin), Berlin, Germany, the Technische Universität Berlin (TU Berlin), Berlin, and Freie Universität Berlin (FU Berlin), Berlin, in 2017. He is currently working toward the Ph.D. degree at the Machine Learning Group, TU Berlin.

Jacob R. Kauffmann received the bachelor's and master's degrees in computer science from Technische Universität Berlin, Berlin, Germany, in 2014 and 2017, respectively, where he is currently working toward the Ph.D. degree with the Machine Learning Group.

Robert A. Vandermeulen received the master's degree in electrical engineering, the master's degree in mathematics, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2012, 2015, and 2016, respectively.

He is currently a Senior Researcher with the Machine Learning Group, Technische Universität Berlin, Berlin, Germany, and the Berlin Institute for the Foundations of Learning and Data (BIFOLD), Berlin.

Grégoire Montavon received the master's degree in communication systems from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2009, and the Ph.D. degree in machine learning from the Technische Universität Berlin, Berlin, Germany, in 2013.

He is currently a Senior Researcher with the Machine Learning Group, Technische Universität Berlin, and the Berlin Institute for the Foundations of Learning and Data (BIFOLD), Berlin. His research interests include explainable machine learning, deep neural networks, and unsupervised learning.

Dr. Montavon is a member of the ELLIS Unit Berlin and an Editorial Board Member of *Pattern Recognition*. He was a recipient of the 2020 Pattern Recognition Best Paper Award.

Wojciech Samek (Member, IEEE) studied computer science at the Humboldt University of Berlin (HU Berlin), Berlin, Germany, from 2004 to 2010. He received the Ph.D. degree (honors) from the Technische Universität Berlin (TU Berlin), Berlin, in 2014.

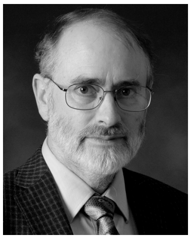
He is currently the Head of the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Berlin. He is also an Associate Faculty with the Berlin Institute for the Foundation of Learning and Data (BIFOLD), Berlin, the ELLIS Unit Berlin, Berlin, and the DFG Graduate School BIOQIC, Berlin.

Dr. Samek is an Editorial Board Member of *Digital Signal Processing*, *PLoS ONE*, *Pattern Recognition*, and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS) and an Elected Member of the IEEE MLSP Technical Committee. He was a recipient of multiple best paper awards, including the 2020 Pattern Recognition Best Paper Award, and a part of the MPEG-7 Part 17 Standardization. He was an organizer of various deep learning workshops. He has been serving as an AC for the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) 2021.

Marius Kloft (Senior Member, IEEE) received the Ph.D. degree from the Technische Universität Berlin (TU Berlin), Berlin, Germany, in 2011, and the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA.

He was an Assistant Professor with Humboldt University of Berlin (HU Berlin), Berlin, from 2014 to 2017, and a Joint Postdoctoral Fellow with the Courant Institute of Mathematical Sciences, New York University, New York, NY, USA, and the Memorial Sloan Kettering Cancer Center, New York. He has been a Professor of computer science and machine learning with the Technische Universität Kaiserslautern, Kaiserslautern, Germany, since 2017. He is interested in the theory and algorithms of statistical machine learning and its applications. His research covers a broad range of topics and applications, where he tries to unify theoretically proven approaches (e.g., based on learning theory) with recent advances (e.g., in deep learning and reinforcement learning). He has been working on, for example, multimodal learning, anomaly detection, extreme classification, adversarial learning for computer security, and explainable AI.

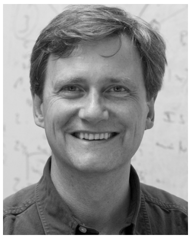
Dr. Kloft was awarded the Google Most Influential Papers Award in 2014. He has been serving as a Senior AC for the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence since 2020 and the International Conference on Artificial Intelligence and Statistics (AISTATS) since 2020. He is also an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS).



Thomas G. Dietterich (Member, IEEE) received the B.A. degree from the Oberlin College, Oberlin, OH, USA, in 1977, the M.S. degree from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1979, and the Ph.D. degree from Stanford University, Stanford, CA, USA, in 1984.

He is currently a Distinguished Professor Emeritus with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA. He is one of the pioneers of the field of machine learning. He has authored more than 200 refereed publications and two books. His current research topics include robust artificial intelligence, robust human–AI systems, and applications in sustainability. He has devoted many years of service to the research community.

Dr. Dietterich is a former President of the Association for the Advancement of Artificial Intelligence and the Founding President of the International Machine Learning Society. His other major roles include an Executive Editor of the *Machine Learning* journal, a Co-Founder of the *Journal for Machine Learning Research*, and the Program Chair of the Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence 1990 and the Conference on Neural Information Processing Systems (NIPS) 2000. He also serves as one of the moderators for the cs.LG category on arXiv.



Klaus-Robert Müller (Member, IEEE) studied physics in Technische Universität Karlsruhe, Karlsruhe, Germany, from 1984 to 1989. He received the Ph.D. degree in computer science from Technische Universität Karlsruhe in 1992.

He has been a Professor of computer science with Technische Universität Berlin (TU Berlin), Berlin, Germany, since 2006. In 2020 and 2021, he is on a sabbatical leave from TU Berlin and with the Brain Team, Google Research, Berlin. He is also directing and co-directing the Berlin Machine Learning Center, Berlin, and the Berlin Big Data Center, Berlin, respectively. After completing a postdoctoral position at GMD FIRST, Berlin, he was a Research Fellow with The University of Tokyo, Tokyo, Japan, from 1994 to 1995. In 1995, he founded the Intelligent Data Analysis Group at GMD-FIRST (later Fraunhofer FIRST) and directed it until 2008. From 1999 to 2006, he was a Professor with the University of Potsdam, Potsdam, Germany. His research interests are intelligent data analysis and machine learning in the sciences (neuroscience (specifically brain–computer interfaces), physics, and chemistry) and in industrial applications.

Dr. Müller was an Elected Member of the German National Academy of Sciences, Leopoldina, in 2012 and the Berlin Brandenburg Academy of Sciences in 2017 and an External Scientific Member of the Max Planck Society in 2017. In 2019 and 2020, he became a Highly Cited Researcher in the cross-disciplinary area. Among others, he was awarded the Olympus Prize for Pattern Recognition in 1999, the SEL Alcatel Communication Award in 2006, the Science Prize of Berlin by the Governing Mayor of Berlin in 2014, the Vodafone Innovations Award in 2017, and the 2020 Best Paper Award in the *Pattern Recognition* journal.